Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2013

Luís Francisco Bento Ferreira

CAMBADA@Home: Deteção e Seguimento de Humanos
CAMBADA@Home: Detection and Tracking of Humans

Luís Francisco Bento Ferreira

CAMBADA@Home: Deteção e Seguimento de Humanos
CAMBADA@Home: Detection and Tracking of Humans

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários a obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicacoes, realizada sob a orientação científica do Professor Doutor António José Ribeiro Neves, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Professor Doutor Artur José Carneiro Pereira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

O Júri

Presidente	**Professor Doutor Armando José Formoso de Pinho** Professor Associado C/ Agregação, Universidade de Aveiro
Vogal - Arguente Principal	**Doutor Agostinho Gil Teixeira Lopes** Investigador Auxiliar, Universidade do Minho
Vogal - Orientador	**Professor Doutor António José Ribeiro Neves** Professor Auxiliar, Universidade de Aveiro
Vogal - Co-Orientador	**Professor Doutor Artur José Carneiro Pereira** Professor Auxiliar, Universidade de Aveiro

agradecimentos / acknowledgements

São muitas as pessoas a quem gostaria de agradecer por me terem ajudado nesta fase da minha vida e na passagem pelo curso de ECT. As mais importantes são sem dúvida os meus pais Luis Carlos e Maria Manuela, mas sem esquecer também os meus avós, padrasto, madrasta, tios, primos e até mesmo o meu irmão de 10 anos que fica admirado quando lhe digo que estou a finalizar um *trabalho de casa* que dura à um ano.

Os anos que passei em Aveiro não seriam o mesmo sem a companhia e apoio da minha namorada Cátia e dos amigos que fiz durante este tempo, e como eles me conhecem bem compreendem que não vou mencionar os nomes de cada um porque seria uma lista grande e provavelmente ia-me esquecer de mencionar alguns, no entanto podem esperar um agradecimento pessoal!

Aos meus orientadores António Neves e Artur Pereira agradeço a oportunidade que me foi dada para trabalhar num projeto do qual gostei bastante, e pela orientação ao longo do seu desenvolvimento, assim como desta dissertação. Não pode faltar também um agradecimento merecido aos colegas da equipa CAMBADA pelo companheirismo demonstrado desde que entrei para a equipa, e em especial ao Eurico e ao Alex pela grande ajuda que me proporcionaram na caça de bugs, pela partilha de conhecimento e pelo bom ambiente no espaço de trabalho.

Um obrigado a todos os que fizeram parte do meu percurso académico.

Palavras-Chave pessoas, identificação, deteção, seguimento, profundidade, cor, trmica, imagem, modelo, correspondência, kinect, histograma, comparação, serviço, robô, pose, localização, estimação

Resumo Este trabalho apresenta uma abordagem ao problema da deteção e seguimento de humanos, usando uma câmara RGB-D. Existem soluções propostas para este tipo de problema, no entanto, algumas são baseadas em técnicas de extração de fundo ou outras e, como tal, necessitam que a câmara se encontre numa posição estacionária. Com o sistema proposto, a deteção e seguimento podem ser desempenhadas enquanto a câmara se move, em tempo real.

O objetivo deste projeto é a implementação de um sistema de deteção e seguimento de pessoas para o robô de serviço CAMBADA@Home, permitindo assim o desenvolvimento de futuras aplicações na área da interação humano-robô.

O sistema aqui descrito permite realizar deteção, classificação e monitorização de múltiplas pessoas. Na primeira etapa, regiões de interesse (ROIs) são segmentadas através da análise do histograma da imagem de profundidade seguido da utilização de um algoritmo de preenchimento. Na etapa seguinte, cada região é classificada como humana ou não-humana através de uma técnica de correspondência de modelos, baseada no algoritmo de descida de gradiantes RPROP, com suporte para múltiplos modelos. A terceira e última etapa permite a monitorização de várias pessoas, através de um método de atribuição de identificadores únicos baseado em comparação de histogramas, assim como estimação de pose e localização.

Os resultados obtidos em ambiente não controlado são encorajadores, com altas taxas de deteção, e, em geral, os algoritmos de estimação de pose e localização são executados como esperado. Para além disto, o projeto CAMBADA@Home foi premiado com o primeiro lugar no Desafio Free Bots, que teve lugar durante o campeonato nacional de robótica, Robótica 2013, onde o robô provou ser capaz de executar rondas autónomas num ambiente desconhecido enquanto detetava e monitorizava pessoas com as quais se cruzava.

Keywords people, identification, detection, tracking, depth, color, thermal ,image, template, matching, kinect, histogram, comparison, service, robot, pose, location, estimation

Abstract This work presents an approach to the people detection and tracking problem, using an RGB-D camera. While there are already solutions for this problem, some are based on background extraction techniques or other, which require the camera to be in a stationary position. With the proposed method, detection and tracking can be performed while the camera is moving, in real time.

The aim of this project is the implementation of a people detection and tracking system for the CAMBADA@Home service robot, enabling the development of further human-robot interaction applications.

The system here described enables object detection, classification and multiple person tracking. In the first stage, regions of interest (ROIs) are segmented through the analysis of the depth image histogram and using a flood fill algorithm. On the next stage, each region is classified as human or not-human using a template matching technique, based on the RPROP gradient descent algorithm, with support for multiple templates. The third and last stage enables the tracking for multiple persons, using a unique identification assignment method based on histogram comparison, as well as pose and location estimation.

The results obtained in unconstrained environments are encouraging, with high detection rates, and, in general, the algorithms for pose and location estimation perform as expected. Furthermore the CAMBADA@Home project has been awarded with the first place in the Free Bots Challenge, which took place on the Robótica 2013 robotics national championship, where the robot was proven to be capable of performing autonomous tours in an unknown environment while at the same time detecting and tracking people it came across.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

What once was fiction is getting to be the reality we live in. Current households are equipped with machines that get smarter everyday and in which we rely to do some of our daily tasks. With the increase in processing and battery performance, and the ever growing capabilities of mobile computers, the logical evolution of social robots is also the increase in mobility and ability for them to understand our requests and needs.

The area of Social Robotics is the study of robots that are able to interact and communicate between themselves, with humans, and with the environment, within the social and cultural structure attached to its role ([4]). For a robot to be able to interact with its surroundings, it has to be able to perform some basic tasks, being one of the most important to see and understands what is being seen. In recent years, there have been many advances in the Computer Vision research area, where some real projects have been deployed and proven to be effective in the interaction between robot and human.

This thesis will describe the work developed in the area of Social Robotics in a recent project, developed at University of Aveiro, named CAMBADA@Home. The aim of this thesis is the development of a system capable of detecting, identifying and tracking humans, using two different types of cameras, RGB-D and thermal, and its implementation in the CAMBADA@Home robot.

This work is deeply connected to the Computer Vision research area and takes advantage of previous work done by researchers such as [5], [6], [7], [8], or [9], who provided many breakthroughs in recent years.

Most of the first works performed in the human detection area only used a color camera to perceive the surroundings, rendering the job very difficult, if not impossible in cluttered environments. Because some of these works were based in background extraction, their application in a mobile platform may not provide the expected results. Some researchers tried different approaches using Stereo cameras, Time of Flight cameras or even Laser Range Finders. Despite the good results achieved from these novel approaches, some are expensive and therefore reserved to selected researchers.

This work aims to create a usable solution for the people detection, identification and tracking problem, in an unconstrained environment applied to a mobile platform, making future works focused in the interaction between robots and humans possible.

1.1 Context and Motivation

The area of social robotics is starting to enter in a more mature stage in recent years, with several developments that allow for robots to enter peoples lives and provide basic services.

As in the history of personal computers, the first machines served only basic general needs. However, with the development of programs that cater individual necessities of a single user, they became an essential tool in peoples lives, controlling their schedule, being the main communication method or serving as work tools. Service robots should see an similar development, providing greater services than current household robots, such as toaster, fridges or more recently smart vacuum cleaners.

The topic of this thesis reflects a basic necessity in human-robot interaction systems, which is the ability to detect and recognize people. A service robot that has these features, has the basic tools that allow for the creation of advance interaction systems.

The main motivation behind this thesis is then the development of a system that, in conjunction with others, allows for a robot to respond to commands given by people, and help them on their daily task. Examples of assistance could be, for example, the guidance and support of elderly people, whose motor capabilities may be diminished, or in an industrial environment, the ability to perform simple tasks such as mail or message deliveries, guiding visitors through a building, among others. Since the area of service robots, capable of human interaction, is still relatively unexplored, new demands may arise with the development of these robots capabilities.

1.2 Objectives

The outline of this thesis can be quickly described as the creation of a system that is able to detect, identify and track humans, however, the implications of each of these features are more complex than their quick description.

The first step is the study of previously developed systems capable of performing these actions. This will aid in the overcoming of issues common to all researchers that use similar methods. Using the knowledge gained from this study, an overall solution should be design, keeping in mind that the one of the goals is its application in a real robot, namely the CAMBADA@Home autonomous service robot, from University of Aveiro. The current system of the CAMBADA@Home robot is being developed using the Robot Operating System (ROS) [10], therefore to achieve greater compatibility, and allow for further development, the support for this architecture is important.

While developing the basic structure of the proposed solution, the use of special cameras, such as a RGB-D and thermal, should be taken into account in order to improve the obtained results.

When the development is concluded, the algorithms should be integrated in the CAMBADA@Home , followed by thorough testing in order to ascertain the performance of the proposed system.

The writing of this dissertation is the last objective and should reflect the presented objectives properly documented.

1.3 Structure for the proposed solution

As stated before, the compatibility with the ROS middleware is an important aspect of the proposed solution, therefore, the system has been divided in several nodes, respecting the ROS ideology. This approach presents several advantages, where the two most important are that each individual node is capable of running in its own process, taking advantage of the full capabilities of the processor; and separating the source of the information (camera drivers) from the Image Analysis node allows for the proposed solution to have a low-coupling.

The diagram in Figure 1.1 presents the main outline of the system, where each node, represented by circles, is capable of publishing or consuming messages, represented by rectangles. The Thermal Preprocessing node could not be complete due to time constraints, as explained in section 3.2, however it is still present in the diagram to exemplify where it would act, but represented by a dashed outline.

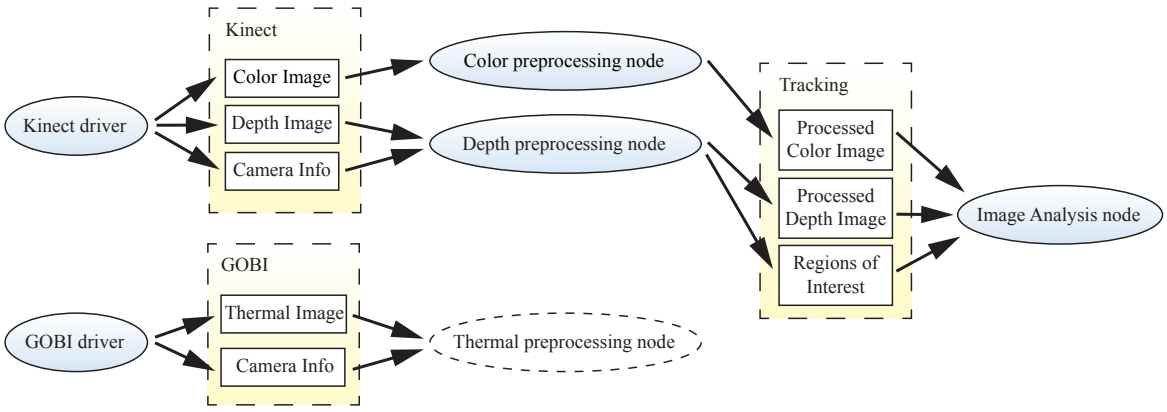


Figure 1.1: Structure of the proposed system.

1.4 Organization of the dissertation

The remaining chapters of this document are organized as follows:

- **chapter 2: State of the art** - Current state of the art for the research areas that contribute to this project. Because this dissertation involves very different and lively areas of research, each one is separated in its own section. The first section focus on the task of people detection and tracking, the different methods used and their performance. The second section describes some of the most recent techniques used for people identification, using features from both color images and thermal images, as well as some machine learning techniques used in the process.

- **chapter 3: Cameras Used** - Description of the image sensors used in the project. The first section describes the main features of the Microsoft Kinect, its specifications and its great contribution for the Computer Vision research area. The second section describes the Xeneth Gobi thermal camera, a forward looking infrared (FLIR) camera, and its specifications. Because there is no driver for this camera that enables its interaction with

the ROS middleware, a sub-section is dedicated to the documentation of the developed module.

- **chapter 4: Obtaining Regions of Interest** - Steps necessary for human-candidate detection. Starting with the procedure of image preprocessing, followed by an explanation of the method used for depth image histogram analysis, image segmentation, which enables the retrieval of ROIs, and finalizing with the filtering of unwanted candidates.

- **chapter 5: Classifying Regions of Interest** - Classification of ROIs retrieved previously, as human-candidate objects. The first section presents a solution for the creation of templates using data obtained by the Kinect camera. The following sections describe the process of template matching and region classification based on multiple information.

- **chapter 6: Human Tracking** - This section is divided in sections that describe each aspect of the tracking. First and second sections deal with the discrimination of multiple users, assigning each one with a unique identifier. Third section presents a solution for pose and global coordinates location. The last section presents a theoretical solution for human identification, since its complete implementation could not be completed due to time constraints.

- **chapter 7: Experimental Results** - Presentation of results obtained with the proposed solution. Results focus more on people detection performance, using different settings, and on two different proposed datasets. The last section performs a commentary on the results as well as brief comparison with the performance of systems proposed by other researchers.

- **chapter 8: Conclusions** - Last chapter of the thesis were a work summary is present, as well as ideas for future work in order to improve the proposed system.

Chapter 2

State of the art

The work developed in this thesis is based on three well defined objectives, people detection, identification and tracking, therefore much of the research done was based on the Computer Vision research area, a very big and lively area where many breakthroughs have been accomplished specially in the past ten years or so. However, Computer Vision is a very abstract field where many research areas meet, namely two of the most important for this dissertation, human object detection and facial recognition. Therefore these two main research areas are divided into their own sections.

Before presenting some of the most recent works in people detection and identification, and since this subject is highly dependent on visions sensors, it is best to present the most used cameras in this field. This is done in section 2.1, where the type of data and the advantages and disadvantages inherent to three different types of cameras, color, thermal and depth cameras, are described. Lastly one presents a more recent hybrid type of cameras, where color and depth cameras are fused in the same device.

Next section 2.2 describes some of the most recent works with focus on the detection of human objects in unconstrained environments, and some forms of tracking. These works are ordered by the type of sensor used instead of the techniques because, despite some of approaches use the same algorithms, they are applied in different types of data and stages of the process. Therefore presenting a complete overview of the entire detection process is more intuitive than describing the usages of the same algorithm in different projects.

The secondary objective of this work is to perform people recognition on the persons detected, where a possibility can be to perform facial recognition. Because this area also has significantly different approaches, section 2.3 presents the most recent and successful, as well as important comparative studies.

2.1 Image sensors used for human detection

The way a robot perceives its surroundings is essential for its understanding of the world. Visions systems are one of the most important sensors in Human-Robot Interaction (HRI) applications (e.g. surveillance systems [11], [12], [13]), and create a bridge between the real world and the virtual world.

The first image sensors used for the task of people detection and identification were *Color image* cameras. These cameras use sensors like Charge Coupled Device (CCD) or Complementary Metal-Oxide-Semiconductor (CMOS) to capture light variations and create color

images. Nowadays these are the simplest and cheapest type of cameras available, and continue to evolve in terms of image quality and resolution, and decrease in physical size and price, making them a popular choice among researchers ([11], [14]).

Due to the method used to capture images these cameras are highly dependent on lighting variations. This is the biggest drawback in the use of this type of cameras, were a picture taken to the same environment with high-level illumination and low-level illumination can alter the perception of the world completely and, because vision systems are normally a source of information for other high-level processes, erroneous readings in this early stage can easily generate greater errors further in the processing pipeline.

The main characteristic of color cameras, as the name implies, is the ability to capture colors and textures from a scene, which is something that depth and thermal cameras cannot do. This is particularly important in skin detection algorithms ([15]) and color gradient analysis ([14]). An example of a color image and two typical industrial color cameras used in research projects are shown in Figure 2.1.

a) **b)** **c)**

Figure 2.1: Example of a capture using a color camera in *a)*, and two color cameras models in *b)* and *c)*.

Thermal cameras, particularly Long-wavelength Infrared (LWIR) cameras, are a popular choice in human detection applications ([16]), because of their ability to obtain a completely passive capture of the world based on thermal emissions, requiring no external light. Infrared light is emitted or absorbed by objects, whether these generate or absorb heat respectively, and the intensity of the pixel in the image will be proportional to the radiation level of the object in that point. This facilitates the segmentation of human objects on thermal imaging because of the difference in radiation levels between people or animals and inanimate cold objects that mainly populate the environment in offices and households. An example of a capture enabled by this type of cameras is shown in Figure 2.2, along with two examples of thermal cameras.

Because LWIR cameras only draw information from radiation levels passively, exterior illumination does not affect the image, encouraging its use in applications such as night time surveillance or environmental monitoring. However, as any other sensors, there are disadvantages such as "low SNR, white-black / hot-cold polarity changes, and halos that appear around very hot or cold objects" [16]. Depending on the environment temperature, the thermal properties of the people and background may present great variations hindering

a) **b)** **c)**

Figure 2.2: Example of a thermal capture in *a)* using cameras such as the ones presented in *b)* and *c)*.

---

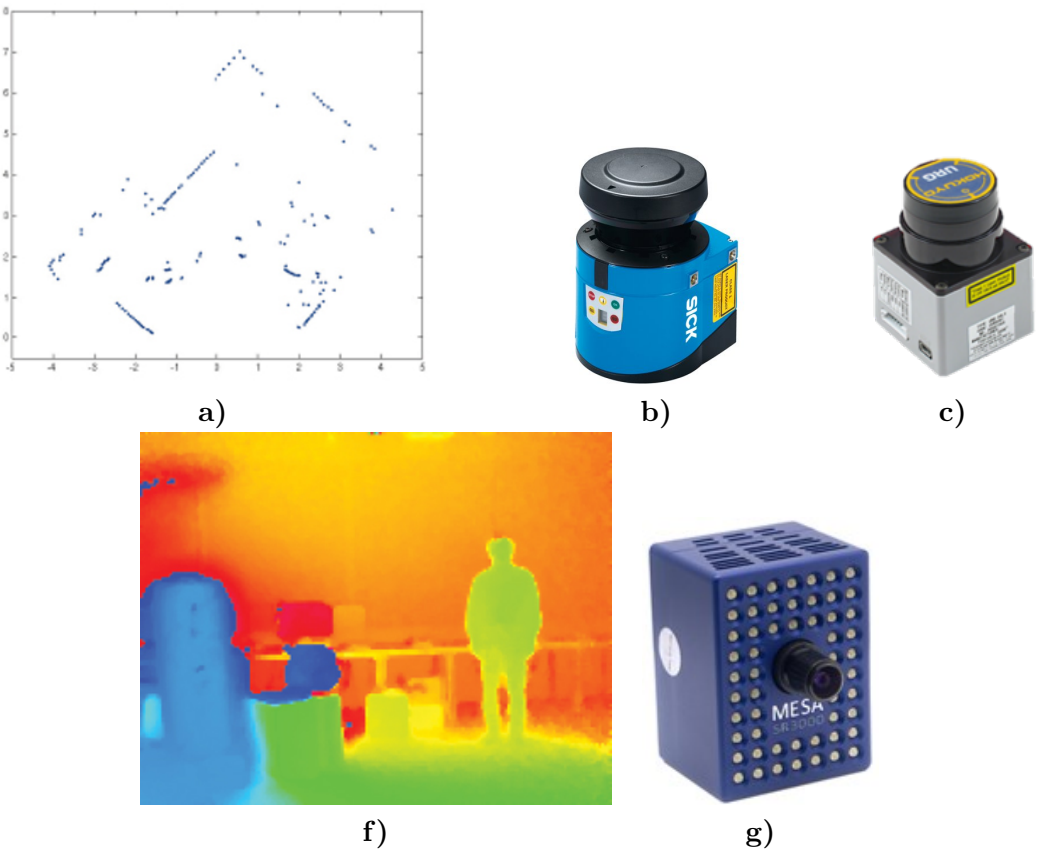
the performance of background-subtraction and template matching techniques.

In the object detection research area the shape of the object is one of the most important visual cues, because "objects may not have consistent color and texture but must occupy an integrated region in space" [9], making *Depth cameras* a popular choice when used along template matching techniques ([9], [11], [4]). Depth cameras are able to perceive the physical world, whose captures can come in three forms:

- 2D Range Data ([17], [18]) - characteristic of sensors that retrieve a single line of points with two dimensions, normally X and Y;
- 3D Range Data or Point Clouds ([19], [6] [8]) - similar to 2D Range Data but instead of retrieving just one line from the surroundings, it retrieves a set of points in three dimensions, X, Y and Z
- Depth images ([20], [4], [9], [21], [1]) - this type of information embeds information on pixels that compose the image. Similar to thermal imaging, the intensity, or color, of the pixels is indicative of a measure, in this case is the distance to the camera.

There are also different types of equipments used to capture this information from the world. *Time-of-Flight* (TOF) cameras area a popular choice due to their effectiveness, availability, and the simplicity of the principle behind their operation: a beam of light is sent from the camera to the surroundings, and the time this beam takes to be reflected and read back by the sensor dictates the depth of that point. 2D and 3D Range data are normally associated with this type of cameras. Figure 2.3 shows an example of a 2D capture in *a)*, taken from [17], where a sensor such as the ones shown in *b)* or *c)*, was positioned at knee height, and a set of points were collect along a 2D plane. It is also possible to see a depth image in *f)*, taken from [6], captured by a device such as the one in *g)*, the real capture does not provide any colors, however to enable its visualization a color map is used.

Another method is known as Stereo Vision in which the images from two slightly separated cameras are correlated, and a disparity image is created. This method is inspired by the human visual system and creates an output in the form of disparity images as the one seen in Figure 2.4 *a)*, taken from [1]. In order to calculate these images any two similar cameras



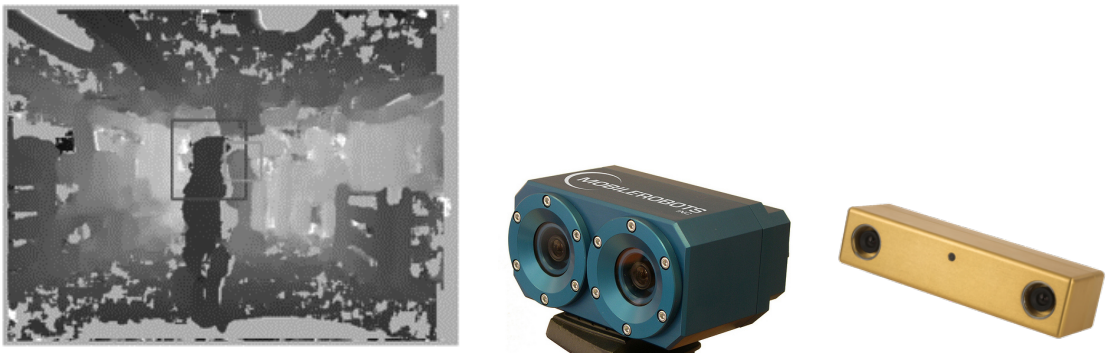
a) **b)** **c)**

f) **g)**

Figure 2.3: 2D range capture from [17] in *a)*, using sensors such as the ones presented in *b)* and *c)*, and a 3D range capture from [6] in *f)*, using the sensor presented in *g)*.

---

can be mounted at a fixed distance from each other, however some commercial solutions are available such as the ones in *b)* and *c)*



a) **b)** **c)**

Figure 2.4: Disparity image from [1] in *a)*, and example of stereo cameras in *b)* and *c)*.

Because these cameras create valuable information about shapes and are lighting invariant,

they represent the main choice in object detection systems, however they lack color and texture information and some can be considered expensive for smaller research projects.

Recently another category of cameras has been created, one that fuses two types of information that complement each other, Color and Depth cameras, named *RGB-D cameras*. These present a solution where the user has access to Color and Depth images (see Figure 2.5 *a)* and *b)*) in the same device, and examples of these cameras can be the very popular Microsoft Kinect or the Asus Xtion (see Figure 2.5 *c)* and *d)* respectively).

Besides the ability to generate both types of images, these cameras have another great advantage: because they were created for a very large audience, casual users and *gamers*, the companies that created them are able to offer such technologies at low prices when compared to most of the systems presented before. The ease of access and the ability to acquire two types of images with the same device made them popular among researchers since the day of release ([9], [8], [3]).

a) **b)** **c)** **d)**

Figure 2.5: Capture of a color and depth image in *a)* and *b)* respectively, using cameras such as the ones in *c* and *d)*.

2.2 People Detection and Tracking

The task of people detection is not a trivial one, if you realize that the human body can assume very different shapes and stances and that a normal household is sometimes cluttered, partially hiding the person. As for the solutions presented for tracking, these vary from location, pose or trajectory estimation.

This section presents some of the most relevant works in this research area ordered by the type of vision system used.

2.2.1 People Detection Using Color Cameras

If we go back to some of the former works developed on people detection ([22], [11], [13]) we can see a tendency to use techniques based on background extraction because they can be applied to any type of images (e.g. color, thermal or depth) and present good results on object detection as long as they fulfil strict requirements (e.g. stationary camera, or a model of the background).

Automated Visual Surveillance Systems have a great impact in security issues and surveillance tasks. In 2000 Haritaoglu, I., et al. [11], designed a system named W4 which was used to track peoples habits (e.g. what are they doing, where and when). This system was developed

keeping in mind a specific environment and hardware. Haritaoglu, I., et al. use a low-cost PC so that the system could be applied in outdoor environments without the risk of large economic losses and intended for large commercial use.

Most of the work on detection and tracking of people using visual spectrum cameras images relies heavily on color cues, for these are important when dealing with texture analysis and represent another dimension of information besides shapes. However outdoors environments and particularly in night-time, to whom this system is aimed, present scenes with low-level illumination where color might not be available. Therefore people like objects need to be detected and tracked based on weaker appearance and motion cues.

The block diagram in Figure 2.6 presents the system architecture for the W4 system. In the first stage Haritaoglu, I., et al. starts by obtaining a background model, which can be accomplished by subtracting each new image from a model of the background scene and thresholding the resulting difference image to determine foreground pixels. This is a simple and common technique that can adapt to slow changes in the scene by recursively updating the model. However modelling the background in outdoors environments presents additional challenges such as small changes in the scene that do not represent a new object (e.g. swaying tree branches). When modelling the pixel value, W4 employs a training period during which the background variation is modelled through bimodal distribution, so that each pixel can be represented by three values: its minimum and maximum intensity values, and the maximum intensity difference between consecutive frames observed during this training period.

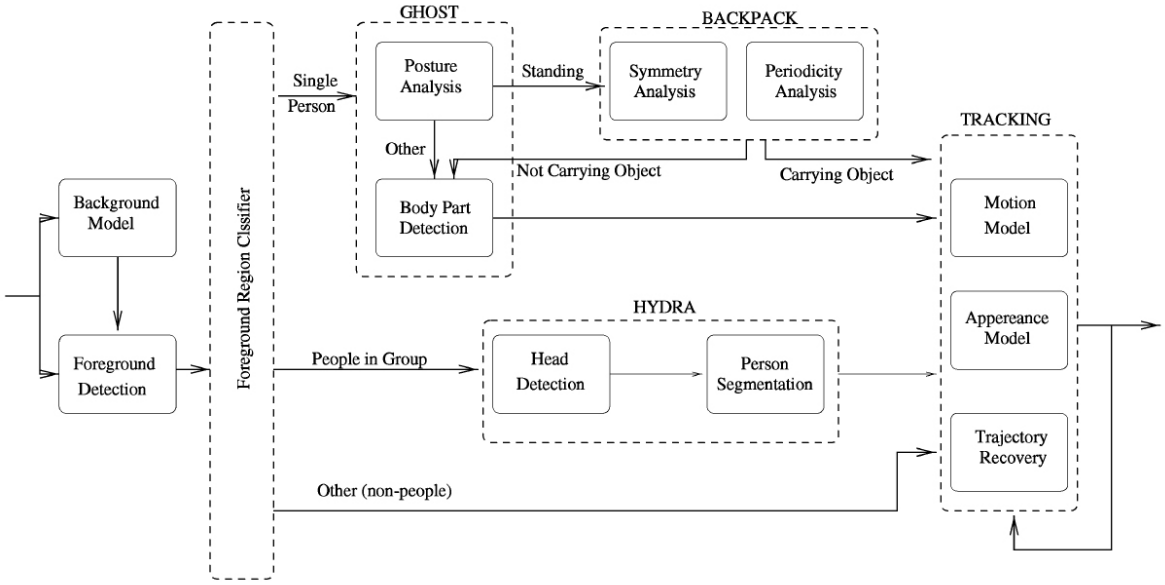


Figure 2.6: W4 System architecture from [11].

During the training period of about 20 to 40 seconds, there are no guaranties that every object in the scene in immobile, therefore a two-stage method had to be applied to exclude moving pixels before the background model computation. In the first stage a pixelwise median filter over time is applied to distinguish moving pixels from stationary pixels, so that in the second stage the initial background model can be construed without interference.

The background model created by Haritaoglu, I., et al. is based on pixel intensity. This

approach is affected by events such as the sun being blocked by clouds, or physical changes when a person exits a parked car. In order to overcome that, the background is updated using two different methods.

A pixel-based update resolves illumination changes in the background, while an object-based update adds objects to the model that are stationary for long periods of time. In order to update the background model a change map is constructed, in which pixels are classified as background or foreground pixels, based on the update methods described before.

To finish the first stage of the process, after foreground regions have been detected, a four stage process is applied in each frame of the video sequence to enhance this detection: thresholding, noise cleaning, morphological filtering, and object detection. To distinguish foreground objects, a binary connected component analysis is applied to label each one. For each labelled region, global and local shape features of the silhouettes are computed. Some of the local features rely on distinct traces of people such as shape, appearance, and motion patterns. These are simple characteristic that can easily distinguish people from other objects in an image. To correlate the same object in different images, W4 uses silhouettes projection histograms and tests their similarity over time.

In the system developed by Haritaoglu, I., et al., there are three predetermined classes: single-person, people in a group, and other objects. In single person detection several problems can take place, like partial occlusions and large changes in the shape of its silhouette even during relatively small movements. These problems cause simple techniques, such as centroid tracking of the foreground blob, to fail. To solve this Haritaoglu, I., et al. applies a two stage matching strategy: in the first stage the displacement of the human object is estimated through the motion of the median coordinate of the person, which quickly narrows the search space for the second stage, where a binary edge correlation between the current and previous silhouette edge profiles is applied.

Besides people detection, surveillance system should be able to recognize someone who has already been in a scene, which can be refereed as people "reappearing" and can happen when a person is occluded by an obstacle or when she leaves and re-enters the field-of-view of the camera. The W4 system combines the gray-scale textural appearance and shape information of person together in a 2D dynamic template called a *textural temporal template* in order to solve this.

In addition to detection and recognition, the W4 system also tries to understand actions performed by the detected people. In order to do this the system tries do predict and track the locations of the six main body parts (head, both hands, both feet and torso). Several silhouette-based body models are compared to the detected foreground object's silhouette and the body posture which yields the highest similarity measure is taken as the estimated posture. Then the location of the head is predict as it is the less deformable part of the body, when compared to other, and can be generally found on the major axis of the silhouette. Next, a recursive convex-hull algorithm is applied to find possible body part locations. When the location of the body parts are known an estimation of the body posture is made, where standing, sitting, crawling/bending, and lying down are main postures, while all other can be seen as variations of these.

To further improve the understating done by the system of the actions performed by the users, the W4 system also detects interaction between human and non-human objects such as "depositing an object (unattended baggage in airports), exchanging bags, or removing an object (theft)" [11]. For example, to detect objects being carried, Haritaoglu, I., et al. estimates regions that systematically violate the symmetry of the body around its body axis,

and through outlier regions (non-symmetric regions) an objected is modelled so that the system can detect "who" is carrying "what".

The work developed by Haritaoglu, I., et al. represents a very complete solution in the People Tracking research area and uses several different techniques to tackle the many obstacles existent in this area.

In 2004 Dalal, N., et. al. [14] state that locally normalized Histogram of Oriented Gradient (HOG) descriptors provide excellent results when compared against existing methods like wavelets ([23], [24]). This descriptor shares some traits with other popular descriptors, such as Edge Orientation Histograms (EOH), Scale Invariant Feature Transform (SIFT) descriptors and shape contexts, however the authors state that HOG descriptors significantly outperform these feature sets.

The algorithm tries to characterize local object appearance and shape using information provided by the distribution of local intensity gradients and edge directions. To achieve this, the image is divided into small spatial regions, also called *cells*, and for each cell an 1-D histogram of gradient direction or edge direction is calculated. After the division in cells a normalization is done to improve invariance to illumination (important in color cameras). This normalization is not done on a cell level independently, instead a measure of local histograms is accumulated creating an *energy*, which is used to normalize each *block*, where *blocks* are formed by several *cells*. These blocks are referred as Histogram of Oriented Gradient (HOG) descriptors and when used with conventional Support Vector Machines (SVM) based window classifier create the human detection chain presented in Figure 2.7

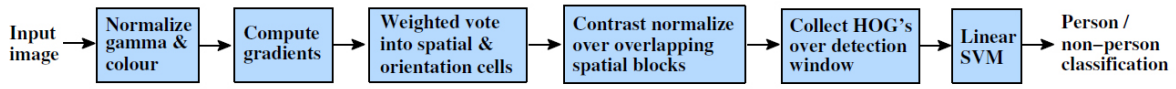


Figure 2.7: Overview of the HOG method [14].

One of the conclusions from this work is that the use of a dense grid, creating in fact overlaps, is essential to the final result, so that each scalar cell response contributes with several components to the final descriptor vector, resulting in a decrease of false positives. The shape of the block was shown to have little effect on the final result, where circular detectors have a slight edge over rectangular detectors, and SVM with Gaussian kernel present only a 3% performance improvement over linear kernel but have much higher run times.

Another important finding is that no blurring should be applied prior to the calculation of the gradient in the hope of reducing sensitivity to spatial positioning. In HOG well defined edges present the main source of image information, and therefore the image should be left in the finest scale available.

To achieve the best results normalization should be applied in each element (edge and cell) several times. A high-quality local contrast normalization is also important.

2.2.2 People Detection Using Depth Cameras

In addition to color cues, the human brain is also capable of perceiving distances. This led researchers to bet in another type of visual information beyond color and texture, namely depth. The advantage of theses systems when compared to conventional color cameras, is that the former are robust to changes in color and illumination and enable spatial object

segmentation because "objects may not have consistent color and texture but must occupy an integrated region in space" [9].

Although presenting good results, methods like HOG features and others recover information from an image through raster scanning. In real-time processing this can present an obstacle when varying the window scale to sizes where the computational cost is too high to achieve a decent frame-rate. Also the use of stereo cameras implies a correspondence calculations between images and stereo matching, which implies added processing costs just for the data acquisition.

This led researchers like Ikemura, S., et al. [6] to propose a method for detection of humans using Relational Depth Similarity Features (RDSF), based on the depth information obtained by TOF cameras, which present depth information without the need for added processing.

In 2010, Ikemura, S., et al. [6] presented a method that uses depth information obtained from a TOF camera to calculate features derived from similarities between depth histograms, that represent the relationship between two local regions. These features are then used to construct a classifier using AdaBoost and allow for fast and highly accurate classification, even when considering occluded regions and people overlapping, achieving a false positive rate of 1% and real-time detection at 10 *fps*. One of the drawbacks is that the used camera, a MESA SR-3100, is incapable of outdoor image acquisition, limiting its use to indoor environments.. An overview of the process can be seen in Figure 2.8.

The algorithm starts by dividing the depth image into *cells* of 8 x 8 pixels. A pair of cells is then selected and a depth histogram is computed for each from the depth information. In order for the cells to be compared a normalization is applied so that the total value of each depth histogram is equal to 1. The result of the comparison from the two cells's histograms is then a "feature that expresses the relative depth relationship between the two regions" [6]. The junction of the degrees of similarity for all combinations of two regions generates a *feature vector*.

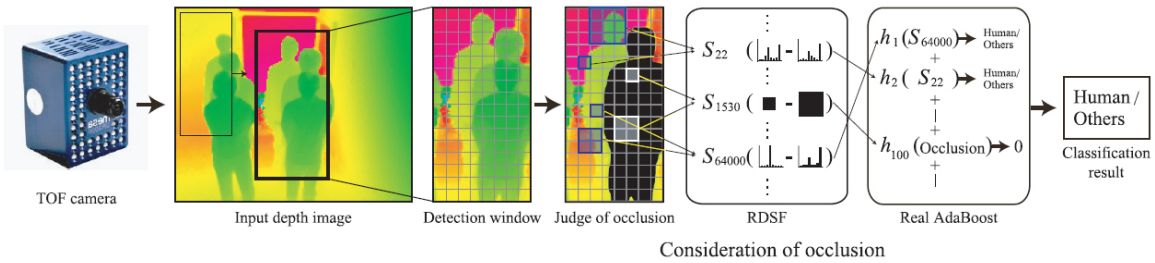


Figure 2.8: Classification process from RDSF [6].

To reduce computational costs in the feature calculation phase Ikemura, S., et al. use integral histograms ("Integral histogram: A fast way to extract histograms in cartesian spaces"). The space is divided by a 0.3 meter spacing, from 0 to 7.5 meters, creating depth histograms formed by 25 bins.

When using such classification methods, features extracted from occluded regions interfere in the correct output of the classifier. To overcome this the output of weak classifiers that perceive this occlusions are discarded. An occlusion is identified as an object region that is at least 0,3 meters closer to the camera than the detection window. For example, when two people are naturally overlapped the depth between them must be at least 0.3 meters to not

be considered an occlusion, and therefore be inserted in different detection windows. This enables the calculation of an Occlusion Rate which can be applied as a weight to the weak classifiers limiting their influence in the final classification.

When collecting the results from the classification process, mean-shift clustering ("Mean shift analysis and applications") is a widely used technique to merge these detection windows into single detection results. However when dealing with 3D spaces, such as the one in this work, errors might occur when classifying humans who occlude other humans. To solve this Ikemura, S., et al. expanded the mean-shift clustering technique from a 2D space into a 3D space, separating clusters also by depth.

Ikemura, S., et al. tested the detection capabilities of the system using HOG features extracted from depth images, RDSFs, and both HOG features and RDSFs. Comparing the RSDFs alone and the mixed features showed that the detection accuracy was almost the same (95%), concluding that RSDFs present the best features during the weak classifier training of the AdaBoost technique.

To tackle the problems of complex background and lighting conditions Arras et al. proposed a method to detect people in two dimensional range scans [17]. The researchers used a Laser Range Finder positioned at leg height to acquire data in a cluttered office environment. At first sight this data appeared to be too complex to be able to distinguish human from other objects such as tables, however the range measurements that correspond to humans presented "geometrical properties such as size, circularity, convexity or compactness" [17].

Knowing this Arras et al. created an algorithm to extract the best features using AdaBoost and created classifiers using these features. The main two types of features used were *geometric* and *motion features*.

In range data, motion features are typically identified by subtracting two subsequent scans, and for the geometric features the data is divided in clusters of points and each clusters is labelled as a line, circle or leg, whereas a leg is a circle with additional parameters conditions. Until the work developed by Arras et al., these parameters were defined manually, therefore this motivated the researcher to implement a learning technique adaptable to any environment.

The boosted algorithm is first trained with a set of training data, labelled positive or negative. In this process several weak classifiers are chosen using a weight distribution, and at each round this weight is increased for the incorrectly classified examples by the previous weak classifier. The final classifier is then composed by a weighted majority of the best weak classifiers.

In order to be classified, these clusters have first to be segmented from the set of beams provided by the laser range finder. Using a segmentation algorithm based on a jump distance condition its possible to create subsets of points, in which beans are grouped in the same subset if they are closer than a certain threshold distance. After the image data is grouped Arras et al. classify fourteen features such as: number of points, standard deviation, jump distance, width, linearity, circularity, radius and mean speed, among others.

To prove that the proposed method was valid Arras et al. tested the system in two different environments, a corridor and an office. According to the AdaBoost algorithm the best five features for both environments are, ordered by importance: the radius of the circle fitted into the segment, the mean angular difference which quantifies the convexity of the segment, the jump distances from the preceding, succeeding segment and the compactness of the segment. Training a classifier with these features allowed Arras et al. to achieve a detection rate over 90%.

Taking the work developed by Arras et al. as a starting point, Mozos, O., et al. took the idea of classifying body parts using supervised learning approach in laser range data even further, by applying this method in several layers (see Figure 2.9) for different body parts (e.g. legs, upper body and head) and creating a final classifier taking into account the classification of all the segments [18].

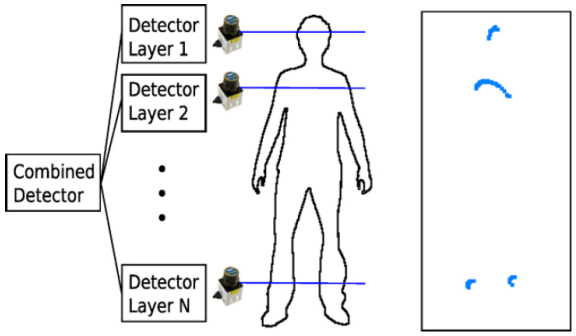


Figure 2.9: Mozos, O., et al. system configuration [18].

The learning method applied in Mozos, O., et al. work is also supervised using the AdaBoost algorithm to create a strong final classifier by combining several weak classifiers. The features chosen to train these classifiers were inspired in the features selected by Arras et al., however, instead of the original fourteen, Mozos, O., et al chose only eleven, discarding the three considered by the researchers as the less important.

In order to combine the output of the different classifiers a probabilistic voting approach is used. First a *Shape Model* of the person specifies the distance relations among the different body parts. To do this first the segments are projected into a 2D horizontal plane, so that a maximum Euclidean distance between segments correspondent to body parts can be measured. This indicates whether two new segments satisfy the distance relation between their corresponding layers. After the conclusion of the segmentation process based on Arras' method , the system determines the probability of a detected segment being a positive example of the body part corresponding to that layer. The final person detection can be achieved by accumulating evidences for all segments found in all layers with an according distribution of probabilistic votes and by selecting the hypothesis with the maximum positive score. Selecting only the maximum positive score allows only for the detection of single human objects, however Mozos, O., et al. state that by looking for different local maxima in the hypotheses space one can detect several persons.

The multi-layer method presents improvements over the single layer detection of the legs, since it can detect different body parts simultaneously and allows for a more robust final classifier combining the different outputs in a probabilistic framework.

2.2.3 People Detection Using Thermal Cameras

Due to its characteristics thermal imaging is a popular choice for human detection systems, such as automated surveillance systems.

In 2005 Davis, J., et al. created a two-stage approach to detect people in thermal imagery [16]. The first stage is use as a fast screening procedure to hypothesize the location of people in the image, and it is necessary due to the problems associated with this type of image, such

as polarity changes in the image (e.g. the person has a higher/lower temperature than the background) and the halos that appear around objects that have very distinct temperatures in relation to the environment. The second stage takes the possible locations of people provided by the first phase and uses AdaBoost classification with adaptive filters to identify the persons in the image.

For the first stage a Contour Saliency Map (CSM) is calculated. This technique creates a map "representing the belief of each pixel belonging to an edge contour of a foreground object (person)" [16] and preserves the input image gradients that are both strong and significantly different from the background (computed using a mean). Therefore the CSM solves the problem of halos and large objects that do not represent persons, as this system is installed at high altitude and the hypothetical detected people have a small area but present a significant difference from the background.

To enable detection of people with different sizes a multi-level CSM is computed from a image pyramid and a generalised CSM template is correlated with the map with multiple resolutions to look for matches. As the person pixels in thermal imagery can vary considerably a fixed template cannot provide the best results, therefore the authors manually extract, normalize and average several cropped windows of people from the CSM image to create this template.

The final stage takes the windows detected by the template matching in the CSM image and determines which ones are persons and non-persons using an assisted learning technique, AdaBoost. The filters used to calculate features in Davis, J., et al. method are sampled from four integral images according to weights provided by AdaBoost in the training phase. This sample is obtained by "finding a subregion in one of the 4 adaptive feature images that gives the lowest weighted error rate" [16], providing an optimal filter for each classifier.

2.2.4 People Detection Using Mixed Cameras

Due to the advantages and disadvantages inherent to each type of system, combining different types of cameras that complement each other is also an usual and sometimes essential approach.

Guan, F., et al. opted for this approach and created a system that uses stereo vision and thermal images to robustly detect human objects [4]. In order to decrease the computational cost of the human segmentation process, the disparity image obtained from the stereo vision system is transferred to a 2D histogram and a scale-adaptive filter is used to extract features of human beings present in the image.

The scale-adaptive filter used presents some constraints as to the human body shape, such as: width, thickness and height. These constraints allow for the exclusion of objects that do not fulfill these requirements, and it acts as the first rejection stage of the algorithm. However, for these constraints to be valid, certain conditions have to be satisfied: the human being should be upright with low inclination, has to be in the field of view of the three cameras and should present an appropriate distance to the camera so that its main features (e.g. head and shoulders) are present in the image.

The implementation of this filter generates a 3D histogram in which the vertical plane represents the belief of the system that a human candidate is present in the image. For each human candidate detected from close to far, a human segmentation algorithm is applied. Finally after the segmentation process is complete, a human verification is carried using a deformable head-shoulder template.

To further enhance the human verification described in the previous paragraph, Guan, F., et al. use a filtered image obtained from a thermal camera to validate human candidates. Using the filter proposed by the researcher it is possible to retrieve from the original thermal image only the pixels associated with a specific thermal feature, such as skin and clothes.

In order to perform the data fusion between the disparity image and the thermal image, the pixels from both images need to be associated. Using equations developed by the author it is possible to transform a 3D coordinate from the depth image to a 2D coordinate in the thermal image.

The system developed by Guan, F., et al. present good results, however the detection through stereo vision alone creates a considerable number of false positives in objects that mimic human shape (e.g. head-shoulder shape). The purpose of the thermal camera in this work is mainly to clarify these cases and when combining the two types of images the system presents a success rate of over 90%.

A different combination of depth image and color image analysis was used by Jain, H, et al. in 2011 to enable upper-body human pose estimation [20]. In this work a model based approach is used for detecting and estimating human pose, using for this purpose a Haar cascade based detection and template matching technique.

While other systems use the thermal camera to prune false positives from the depth detection stage, Jain, H, et al. apply the following detection logic: first the system segments the foreground region to be analysed, next a Haar feature detector is applied and a form of AdaBoost learning is used to train a classifier for upper-body (head+torso) detection, a frontal face detection classifier is applied and, if it fails, a profile face detector is applied, in case both of the classifiers fail then the region is assumed to be a false positive for the upper-body detection. This method assumes that, for a upper-body to be detected, it has to contain either a frontal or a profile face, or else a bad detection is performed.

The form of AdaBoost used by Jain, H, et al. is organized as a rejection cascade of nodes, where each node is a multi-tree AdaBoosted classifier, which allow for faster detection.

To reduce the computational cost associated with template-based matching techniques, the number of sampling points is reduce by downsampling both the search and template images by the same factor. The templates for the matching are obtained by successful detections in the last frame providing a more accurate template. The Haar cascade based detection is only used when the detection is being performed for the first time or after a chosen time-lapse to handle drifting errors of the template and the appearance of other human subjects in the image.

After a successful detection and segmentation comes the fitting process for the stick human body model with 7 body parts, as this is the simplest representation of the human body in form of data. The head, neck, shoulder (both left and right) joints are estimated and fitted based on the detection performed in the template matching process, while the rest of the joints (elbow and wrist) are fitted using a linear regression on sampled weighted-distance transform map. Using Distance Transform (DT) each pixel is mapped according to the smallest distance to a region of interest.

The balanced use of Haar cascades, for detection and drifting errors in tracking, and template matching for tracking variations in object pose in a computationally light approach, allows for the system to perform in real-time with good results.

After conducting several important studies in facial recognition using thermal imaging, Correa, M., et al. developed a complete system responsible for the detection and identification of humans [7]. For this task the researchers use images captured from the thermal and visual

spectrum. Using both thermal and RGB images a skin detection and human detection module are employed in order to detect blobs, that are further analysed in an *Integrated Blob and Detection Analysis module*.

The human skin detection module works in the visual spectrum using the *Skindiff* skin segmentation algorithms, and on the thermal spectrum by means of Mixture of Gaussian (MoG) parametric probability model of the distribution of temperature of skin. The *Skindiff* is a two-stage fast skin detection algorithm, where the first stage performs a pixel-wise classification using a non-parametric skin model implemented using histograms, and the second stage that takes neighbourhood information into account when classifying a pixel, starting by those that have a large likelihood of being a skin pixel. The use of MoG in the thermal spectrum enables the modelling of skin and non-skin pixels through probabilistic distributions, using a Bayes classifier and skin probability model obtained using a training database.

The results obtained from the human detection modules are then integrated in order to be analysed by a Face Detection module. This module uses a multi-resolution analysis approach using boosted cascade classifiers that classify a sample window as containing or not a face. Real AdaBoost is used in each layer of the cascade classifier in order to allow "higher classification accuracy and processing speed by reusing information in each layer the confidence given by its predecessor" [7].

For face recognition in frontal faces candidates Correa, et al. selected the histograms of LBP features as the best methodology for face recognition based on the studies provided by [25]. This method proved to be robust under variable illumination and view angles and up to a distance of 6 meters, making it appropriate for domestic applications.

A different approach, in which the detection of humans is based on a combination of 3D depth and 2D image data, was developed by Hegger, F., et al.. The researchers use data provided by a Microsoft Kinect in the form of 3D point clouds and a horizontal layer division is applied creating smaller data clusters. These clusters are then analysed and a novel feature vector is design using LSN and statistical features.

The choice of using histogram of local surface normals (HLSN) as a feature vector is due to its properties representation. In normal households a reasonable part of daily environments consists of horizontal and vertical planes (e.g. walls, pillars, tables, chairs), whereas the human body has a more cylindrical appearance.

The choice for the classifier was made taking into account tests made for this specific feature vector, where Random Forest (RF) achieved the best results when compared to AdaBost and SVM. Because the RF algorithm expects an one dimensional input vector, the featured vector had to be separated in a histogram for each axis (x, y and z), and also the width and depth of the cluster is added as an extra feature to decrease the false positive rate.

The final step is to assemble the clusters that are classified as human. To do this Hegger, F., et al. use a graph-based representation based on the clusters center. A successful detection has to contain at least three clusters, demanding that at least 45 cm of the person's body must be visible.

Tests on the system revealed it is robust, even with a high level of occlusion. The training was performed with persons standing, which is noticeable in the results for the detection of this type of detections when compared to sitting and partially occluded cases. The detections of people sitting present a decreased detection rate because the classifier was not trained for this events.

Inspired by the the popular featured detector, HOG [14], Spinello, L., et al. took a different approach while creating his Histogram of Oriented Depths (HOG) for people detection [3]

using a RGB-D camera, namely Microsoft Kinect. While HOG is a widely used technique for featured detection in visual images, HOD represents a similar method but to be used with dense depth data.

The main characteristics of the HOG method are shared by the novel HOD method. A fixed-size detection window is defined and divided in an uniform grid of overlapping *cells*. In each cell a descriptor is computed and the *oriented depth gradients* are collect into an single dimension histogram, due to dimensionality limitations of the SVM learning algorithm. As in HOG, *blocks* are defined as a set of cells and are used to normalize the histograms, which increases the robustness of the final classification even under noisy data.

The resulting set of HOD features are then fed to a SVM to enable training and later classification.

Due to the nature of the information collected in these features, silhouette blocks at the contour of objects represent the main source of information, as confirmed by higher weights assigned by the SVM algorithm. This makes image preprocessing an important step preceding the feature collection, even more considering the Microsoft Kinect's low precision at high distances. For this reason Spinello, L., et al. apply pre-processing in the raw range image that resembles the effect of gamma correction in visual images, where the image contrast is enhanced, but in this case the function works on the depth values.

Another feature of this algorithm is an improved searching method. In algorithms, such as HOG, a scale-space search (e.g. image pyramids) is used to find objects in the image from which features are going to be calculated. This is a computationally demanding procedure and so what the researcher proposes is an *informed scale-space search* in which only search windows compatible with a predetermined scale S, calculated using variables such as the average height of a person and the depth of the pixel, are allowed to be forwarded to the SVM. This method avoids the consideration of many scales during the search phase, which translates in an optimized process whose performance is almost the triple of uninformed search heuristics.

Taking advantage on the use of a RGB-D camera, which is capable of creating visual and depth images, and the similarities between the two algorithms, Spinello, L., et al. now proposes a combination of the two feature descriptors. Depth images are robust to illumination changes but sensitive to low-signal strength and have a limited depth resolution, whereas visual spectrum images are rich in color and texture but suffer greatly from variances in illumination. Training a HOG detector on image data and HOD on depth data and using the information from both classifiers with balanced weights presents a novel and robust method for people detection using multi-modality, and where single-cue detectors would fail.

Tests show that the combined classifier outperforms the individual classifiers, and, more important, it enables detection of up to eight meters, whereas the manufacturer's recommended distance is set to approximately 2.5 meters.

2.3 People Identification

Face information is by far the most used visual cue employed by humans when trying to identify one another. Therefore facial recognition is a crucial task for Human-Robot Interaction (HRI) interfaces, such as the one being developed in this dissertation.

Nowadays, computational face analysis is a very lively and expanding research field, making use of different technologies, such as RGB cameras and Long-Wave Infrared (LWIR)

cameras, and different techniques, from classical Eigenspace-based methods (e.g. eigenfaces [26]) to sophisticated systems based on high-resolution images or 3D models. The biggest obstacles this area faces are the variable illumination conditions affecting visible spectrum cameras and variable face expressions.

Eigenspace-based methods, like Eigenfaces, are among the first and most successful methodologies for facial recognition in digital images. First presented in 1991 by Turk, M., et al. [26], Eigenfaces has several variants, most with a real impact in this type of research.

Ruiz-del-Solar, J., et al. [27] provide a comparative study on eigenspace-based methods for computational recognition of faces, along with basic explanations on them. The covered methods are: PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), LFA (Local Feature Analysis), ICA (Independent Component Analysis), KPCA (Kernel PCA), and KFLD (Kernel Fisher Linear Discriminant). The researcher present an independent comparative study and concludes that, considering recognition rates, generalization ability and processing time, the best results were obtained with the post-differential approach, using either a Bayesian Classifier or SVM. Furthermore kernel methods obtained the best recognition rates but also suffered from problems such as low processing speed and the difficulty to adjust the kernel parameters.

Ruiz-del-Solar, J., et al. [25] performed a very complete study on face-recognition methods using visual spectrum images. Ruiz-del-Solar, J., et al. compared some of the best algorithms considering their performance in former comparative studies, in addition to be real-time, to require just one image per person and to be fully online. Two local-matching methods, Histograms of Local Binary Patterns (LBP) features and Gabor Jet Descriptors, one holistic method, generalized Principal Component Analysis) (PCA), and two image-matching methods, SIFT-based and Extremely Randomized Clustering Forest-based (ERCF), were analysed and compared using several databases such as the FERET, LFW, UCHFaceHRI, and FRGC databases. LBP-based methods and Gabor-based methods present the best selection in real-time operation as well as high recognition rates, however Gabor-based methods are slower than LBP ones. PCA methods got the worst results, and SD methods where showed to have a large dependence to illuminations conditions.

In an important comparative study on face-recognition methods for HRI applications using thermal infrared images, Hermosilla, G. [28] compared three algorithms, taking in consideration their suitability for HRI use and their performance from former comparative studies. In this survey the chosen algorithms respect requirements such as online and real-time operation, one image per person and unconstrained environments, resulting in the selection of the following methods: Local Binary Patterns (LBP) Histograms, Gabor Jet Descriptors and Scale-Invariant Feature Transform (SIFT) Descriptors. Although the same method did not always obtain the best results in each test, LBP Histograms presented the overall best result for HRI applications.

After the study presented in the previous paragraph, Hermosilla, G., et al. continued the research in facial recognition in infrared images using a novel method. Hermosilla, G., et al. studied the effectiveness of vascular networks for recognition purposes in [29], using a standard wide baseline matching methodology for the first time. Vascular networks were obtained through skin segmentation and morphological operators applied in the thermal image, and the image matching process makes use of SIFT descriptors employed by classifiers. The work proved that vascular network images preserve important discriminative information about the original thermal images and are robust to variable face expressions.

Chapter 3

Cameras Used

This chapter presents an overview of the cameras used, their advantages, disadvantages and contributions to this work.

The Microsoft Kinect, presented in section 3.1, is a camera that plays a crucial part in the detection and classification of persons for this project, taking advantage of the depth and color images captured by it. Another camera used is the Xenics Gobi-384, presented in section 3.2; this is a thermal camera capable of capturing images of heat radiated by objects.

Some researchers, such as [4] or [7], use a combination of different cameras due to their capabilities of compensating each other's disadvantages. Guan, F. et, al. uses a combination of stereo and thermal cameras, while Correa, M., et. al. prefers a combination of a color and thermal camera.

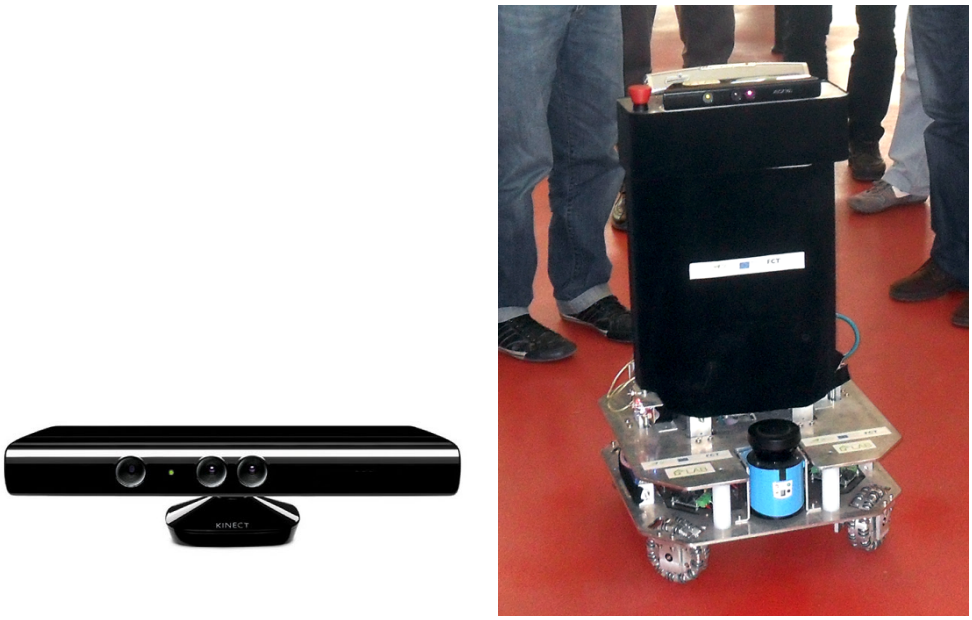
Object detection in color images, for objects with several colors, is a difficult task, while in the depth image it is a more trivial task if the object's shape is not too complex. The advantage of using thermal imaging is that, only certain objects radiate heat, such as human objects or electronic appliances, therefore the thermal camera would work as a powerful confirmation tool for human classification. However, due to time constrains the use of this camera could not be completed, as detailed ahead.

3.1 Microsoft Kinect Camera

Just as computer games were the main driving force behind the increase of graphical processors performance - giving birth to GPUs more capable than some CPUs - they were also the main reason why Microsoft invested in a technology to create "human controllers" for their games. This technology is known as Microsoft Kinect, and even before its official release, exclusively to the XBox 360 gaming console, many researchers understood what it could do for the Computer Vision research area due to its low cost compared to other cameras with similar capabilities.

The joint efforts of the OpenNI community resulted in the creation of a hacked driver to control the communication between this camera and the PC, just only after one week of its public release. After giving up from going against the request of millions of users, Microsoft officially released its own official driver and IDE for Kinect for the PC platform, making it possible for non-expert users to experiment with 3D vision beyond games.

The Kinect sensor (see Figure 3.1 *a)*) is equipped with two cameras and a multi-array microphone. This equipment is disposed in a horizontal bar, that is connect to a motorized



a) **b)**

Figure 3.1: Microsoft Kinect sensor in *a)*, and the CAMBADA@Home robot in *b)*, with the Kinect on top.

---

pivot enabling it to tilt up and down. From the left, the first lens covers a projector responsible for emitting a grid of infra-red points that are going to reflect in the surface of objects, to be later collected by the third lens; the second lens is responsible for capturing color images. In Figure 3.1 *b)* it is possible to see the adapted Kinect camera on top of the CAMBADA@Home robot.

The color image captured by the device has an 8-bit precision and VGA resolution (640×480 pixels), and it is capable of capturing images at 30 Hz. The size of the image is not very good, taking into account the current state of the art for small sized cameras. However the main feature of the Microsoft Kinect is its ability to create depth images, and therefore the color image is normally used as a complement.

The depth image also has a resolution of 640×480 pixels, 16-bit precision, and capturing frequency of 30 Hz. Most 3D spacial cameras are based on a "time-of-flight" technique, in which infra-red light, or an equivalent invisible light, is sent out into a 3D space, and the time it takes to reflect on an object and return to the lens, determines that objects's distance. Microsoft Kinect creators, PrimeSense, use a different approach in which information is encoded in light patterns (see Figure 3.2), and the deformation suffered by those patterns when project onto surfaces, is able to be read and quantified in numbers [30]. These numbers form a 16-bit depth array, but they do not immediately represent the distance of a point relatively to the Kinect. To obtain a depth array where each position of the array holds the distance in millimetres to the camera, a supervised learning method is employed in order to relate values to distances.

Household environments tend to be complex and populated by many objects and colors, making the task of object segmentation on color images complicated. One of the main ad-

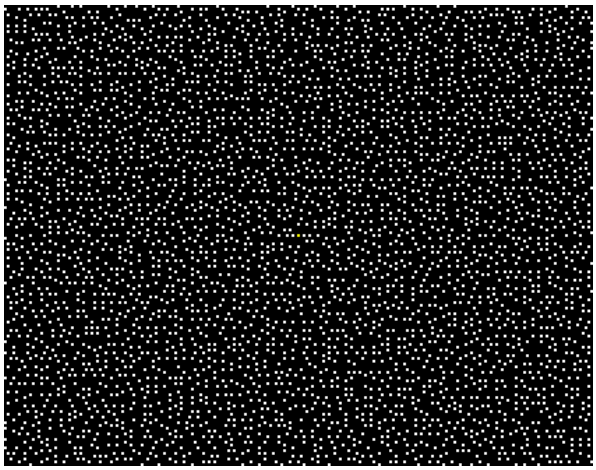
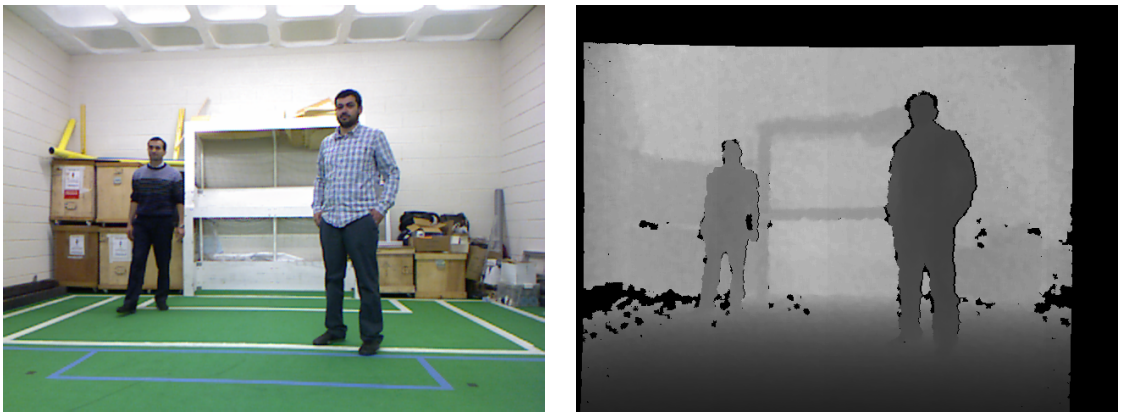


Figure 3.2: Microsoft Kinect infra-red light pattern [2].

vantages of the Kinect is performing this segmentation on a depth image, which proves to be less complex than in a color image, as surfaces can be seen as continuous areas that share the same depth value. In the system developed for this thesis the segmentation of human candidate objects is performed resorting only to depth images.

There are, however, some disadvantages inherent to this type of captures. First, the Kinect has a limited view distance when it comes to the depth image. Throughout several tests, the furthest distance that the Kinect was able to measure was 9757 millimetres, with points beyond this distance being represented as zeros in the depth array, and known as discarded pixels. Furthermore the precision of the Kinect lowers considerably with the increase in distance due to quantization, where objects beyond approximately three meters start to present irregular contours, and some materials (such as glass, hair, or some reflective surfaces) deform the reflection of the project pattern, inhibiting the Kinect to read it, generating discarded pixels in the depth array. Also its field-of-view is very limited.



a) **b)**

Figure 3.3: Example of color and depth captures from the Kinect.

An example of a capture is shown in Figure 3.3, in *a)* the color image, and in *b)* the

depth image. The depth image presented was converted from a 16-bits encoding to 8-bits, as explained in section 4.1, and registered into the color image, hence the black border that can be seen in the top and right limits of the image, and also throughout the examples of depth images presented in this thesis. Notice how in the depth image some patches are composed by discarded pixels.

3.2 Xenics Gobi Thermal Camera

The Xenics Gobi-384 is a Long-wavelength infrared (LWIR) camera used to capture thermal images. The advantage of using a thermal camera is that it is capable of detecting objects that radiate heat, such as persons, and it is robust to changes in lighting. However, some electronic appliances also generate heat and are often present in a normal household (such as a television or lamps), generating false positives in human detection.

The image provided by this camera has a resolution of 384 $\times$ 384 pixels, with 16-bit encoding, and presents a sensitivity for wavelenghts between 8 - 14 μm, with a frame rate of 50 Hz [31].

Figure 3.4: Xenics Gobi thermal camera.

As in projects from other researchers, this one would also benefit from the use of both, a thermal camera and a RGB-D camera, however, due to time constraints the use of the thermal camera was very limited. Nonetheless, a study was made as to the advantages of using a thermal camera to perform facial recognition and as an extra stage of confirmation for human-candidate object classification.

No driver is available for this camera for the ROS platform, however, because one of the objectives of this work was the development of a system fully compatible with the ROS architecture, a module was developed to perform the publishing of images using the ROS middleware.

3.2.1 Thermal Camera ROS Driver

In order to create a driver compatible with the ROS framework, the original driver had to be adapted. Using low-level functions provided by the original driver, the ROS driver is able to acquire images captured by the camera.

The images captured by the camera can be configured by the driver, using several types of filters and color maps. For this project a auto-gain filter is applied and a grayscale color profile is used to visualize the image. As with the depth images captured by the Kinect, the thermal images used by the proposed system have to be converted from 16-bit to 8-bit images, to enable their use in later stages of the system. Figure 3.5 presents a simple diagram explaining the process behind the capture and publishing of images in the ROS environment.

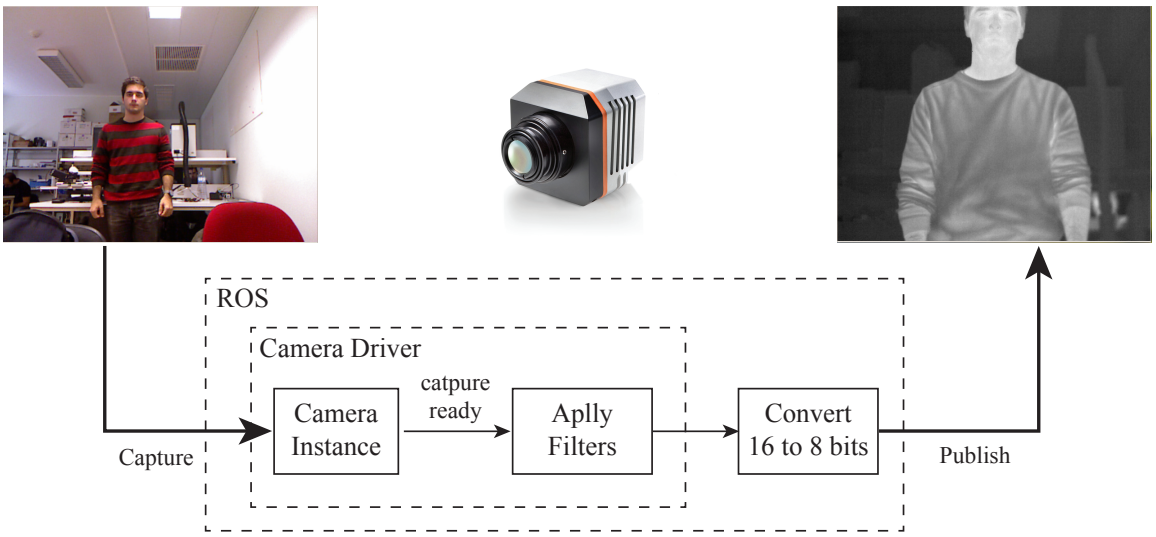


Figure 3.5: Operation diagram for the thermal camera driver for ROS.

Using the developed driver some images were captured and after an analysis some conclusions were drawn. Observing Figure 3.6 it is possible to see that the area captured by the GOBI camera is much smaller than the one capture by the Kinect camera, meaning that, if the camera was used to perform validation for a human object classifier, it could only be used if the object was in the area shared by the two images, which would be small.

Another drawback of using this camera for people detection is that its lens only provides manual focus. Observe Figure 3.7 where in *a)* the lens focus was adjusted for close captures, and in the following images *b)* and *c)*, this focus was maintained while the person moved away from the camera. It can be seen that the camera easily to blurs objects that are not at the correct focus distance.

This effect can be harmful for facial recognition systems, such as the one presented by Hermosilla, G, et. al. [29], and would require the robot on which the camera is mounted, to perform active engagement to correct the distance between the person and the camera.

a) **b)**

Figure 3.6: Capture of a color image and a thermal image.

a) **b)** **c)**

Figure 3.7: Example of several thermal captures at different distances with the same lens focus level.

Chapter 4

Obtaining Regions of Interest

This chapter explains the process relative to the object detection capabilities of the system. In order to detect regions of interest (ROIs), an image analysis is performed over the depth image acquired by the Microsoft Kinect camera. The people detection algorithm relies only on depth information, making this type of data the most relevant for this stage, and therefore presents the most complex preprocessing of the two types of images used.

The detection algorithm starts by converting the depth image, followed by the calculation of its histogram, in order to create slices of the most important bins. Due to the nature of the image, the most important bins are related to the most physically occupied zones in the field of vision of the Kinect, and hence the most probable places for a person to be located. After determining these most relevant levels, the image is sliced in several perpendicular planes relative to the Kinect, facilitating their individual processing and attainment of human candidate regions, also denoted as ROIs, through image processing techniques. The obtained ROIs are then analysed and classified through the methods described in chapter 5. This process is presented in Figure 4.1.

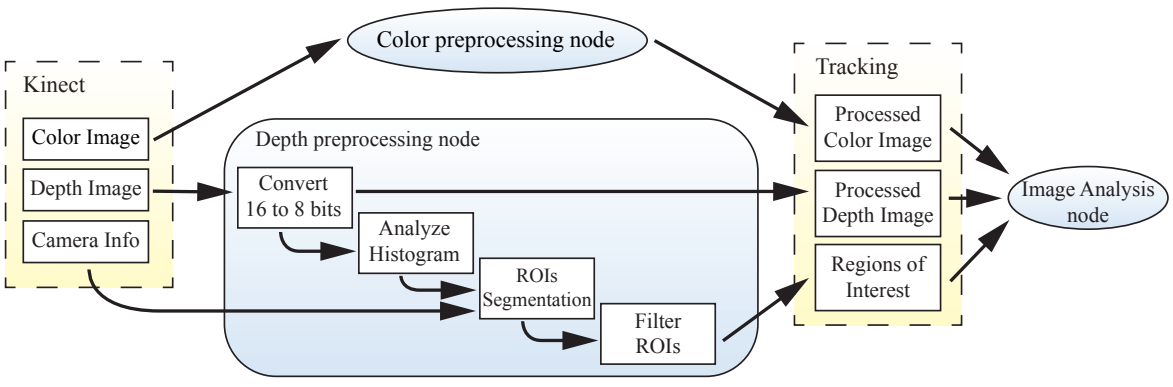


Figure 4.1: Diagram detailing the image preprocessing stage and ROI segmentation.

4.1 Image Pre-processing

In order to improve the independency of the overall system from the cameras used, and also to enable the development of a solution with low coupling, individual ROS nodes were created to perform the necessary preprocessing of the images. This approach enables the separation of the driver nodes that communicate with the cameras from the nodes that appear in later stages of the pipeline. Because of this the rest of the processing pipeline from this point forward is camera independent as long as it receives the data in the expected format.

As mentioned before the color image does not suffer any type of image processing, it is simply bypassed from the camera driver node to the next stage in the human detection and tracking pipeline. The expected encoding for the color image, in this case provided by the Microsoft Kinect, is an 8-bit RGB image.

The depth image provided by the Kinect ROS driver comes in the form of a 16-bit depth array of 640 rows by 480 columns, where each cell stores the depth value in millimetres of that point in the real world. Because this project was developed using the OpenCV framework [32], some of the image-processing methods do not support matrices with encoding larger than 8-bits, and because one of the aims of this work is real-time operation, time constraints are involved. Therefore, the preprocessing node for the depth image is in charge of converting the 16-bits depth array to an 8-bit gray-scale image in order to reduce computational cost associated with the following phases of the pipeline and enable the use of built-in OpenCV methods to reduce development time.

When downgrading a value from a 16-bit precision (65536 possible values), to an 8-bit precision (256 possible values) the loss of information in unavoidable. However not all the values in the original range have the same importance for the algorithm and therefore some can be discarded allowing for the most important to maintain the best possible precision. Furthermore, as explained in chapter 3, although the encoding of the image allows for 65536 different values, throughout several experiments the highest depth value that the depth array stored was 9757, meaning the Kinect cannot see beyond 9.7 meters of distance approximately. Due to the small vertical field-of-view of the Kinect, close objects tend to go outside of the captured image, and due to the interpolation of distant points discussed in section 3.1 far away points present incorrect measures, causing these points not to carry useful information for the template matching process performed ahead.

There are several options when converting the original image to one with smaller encoding: the image can be clipped for the 256 available values in the 8-bits, it can be scaled fitting all the original values in the final range, or lastly a scaling can be done but using only an dynamic range from the original image.

The first option clearly is not suited to our needs, because clipping the original values into a range of only 256 values mean that only a slice of 0.256 meters of depth of the environment would be captured, and this is barely enough for a person to be completely segmented. Scaling the total range into 256 values is a better option than the previous. However, even if the original image is considered to have only 9757 true values of the possible 65536, this still means a great loss of precision. The last option is to perform a scaling of the dynamic range that has the greatest importance for this project's objectives and discard values that are outside this range, since they are less important.

As described in chapter 5, the classification of humans relies on head and shoulder detection, hence human objects closer than 1.0 meter to the Kinect, in general, will not have this area of the body visible due to the height of the camera mounted on the *CAMBADA@Home*

platform and due to the vertical field of view of the Kinect (see Figure 4.2 *b)*). For the same reason, visibility of the head and shoulders of the persons, human objects captured far away from the Kinect have a tendency to present very irregular contours (see Figure 4.2 *c)*), effect that is aggravated if the environment is illuminated by natural lighting.

a) **b)** **c)**

Figure 4.2: Depth images obtained from the Kinect camera. In a) proper full body capture, b) head partially outside of capture, c) irregular shape at a distance, missing the subject's head.

The larger the desired range, the less precision the final the converted image will have but the further the system will be able to detect a person. Because the system classifies human objects by their shapes, the view distance is more important than the precision of the values that compose that object and therefore the range of interest used in this project is situated between the 1.0 and 9.0 meters.

The conversion of the depth array using a dynamic range is done using Equation 4.1, where $\mathcal{I}_c$ is the depth array representing the converted image, and $\mathcal{I}_o$ the original depth array with the same size. u and v are used as indexing variables for the rows and columns of the arrays, respectively. Variables ψ and γ represent the minimum and maximum depth values in meters respectively, and b the number of bins of the histogram, which in this case is equal to 256 in order to take advantage of the full 8-bit precision.

$$\mathcal{I}_c(u,v) = \begin{cases} b \times \left(\frac{\mathcal{I}_o(u,v) - \psi \times 1000}{(\gamma - \psi) \times 1000}\right) & \text{, if } \psi < \mathcal{I}_o(u,v) < \gamma \\ 0 & \text{, otherwise} \end{cases} \tag{4.1}$$

The result of this conversion is an image where low brightness values (dark) represent close points and high brightness values (bright) represent far away points, while areas painted in black (brightness level 0) represent discarded pixels, either due to the incapacity of the Kinect to calculate their depth or due to their presence outside the range of interest.

As in the system proposed by Spinello, L., et. al [3], the proposed system is able to perform detection in a range of almost four times the one recommended as the *adequate play space* in the Kinect User Manual. Figure 4.3 was taken from [3], and shows, in the blue line, the function that relates the byte values of the range image to metric depth. The green area delimits the *adequate play space* for Kinect games, and the yellow area the detection range for Spinello, L., et. al's system. The red dashed line shows the sensors minimal measurable depth, at approximately 0.4 meters.

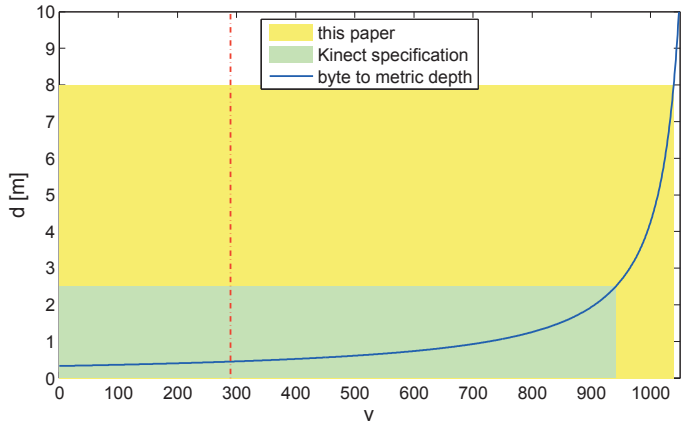


Figure 4.3: Function that relates the byte values of the range image and metric depth, shown in the blue line, from [3].

4.2 Depth Image Histogram Analysis

One of the advantages of working with depth images instead of visible spectrum images, is that the former capture physical objects instead of colors, which makes the process of segmenting objects in a cluttered environment simpler when compared to color images. Furthermore, because the intensity of each pixel is associated with a distance, human objects present smooth intensity transitions which enables the use of known contour detection techniques.

Pixels with similar value have a high probability of belonging to the same object, which enables a histogram based analysis of the depth image, in which the most occupied bins represent possible distances where persons are located. By determining these distances the environment can be divided in vertical slices, from a starting depth to and ending depth, which enables easier segmentation of objects even in cluttered scenes.

In captures like the one seen in Figure 4.2 *c)* the discarded pixels should not be considered a valid object, despite occupying a considerable portion of the image and having the same intensity value. Therefore bin of intensity 0 should be ignored on the histogram analysis performed ahead.

4.2.1 Determine Local Maximums

The first step to obtain the regions mostly occupied in the scene, is to detect the local maximums of the histogram, meaning places where there are larger concentration of points. However, due to the conversion made in the image preprocessing stage and the interpolation performed natively by the Kinect for distant points, bins higher than a certain value start to suffer from high variations creating improper local maximums. Therefore, before the detection of the histogram maximums, a median filter is applied only to the bins whose count is equal to 0 to smooth the graphic, using equation Equation 4.2:

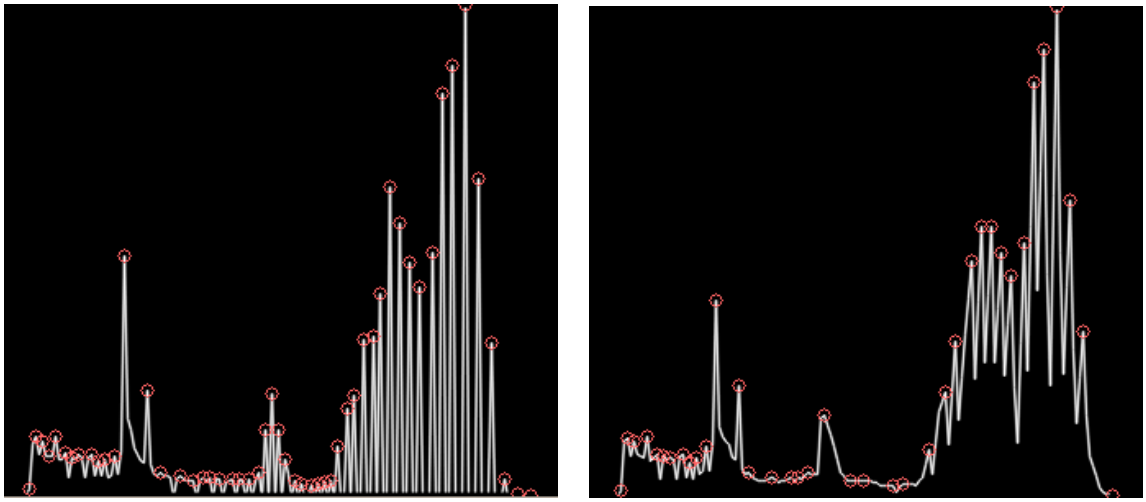
$$\mathcal{H}(i) = \begin{cases} \left(\frac{\mathcal{T}(i-1)+\mathcal{T}(i+1)}{2}\right) & \text{, if } \mathcal{T}(i) = 0 \\ \mathcal{T}(i) & \text{, if } \mathcal{T}(i) > 0 \end{cases} \text{, where } 0 \leq i \leq b \tag{4.2}$$

In Equation 4.2 $\mathcal{T}$ is the image's histogram and i the number of the bin, where its value can go from 1 to 255, which was defined by b previously.

With the histogram now more balanced, with a stronger effect for distant points, it is possible to detect the bins that represent local maximums more accurately. Let $\mathcal{F}$ be a set composed by the indexes i of the bins who respect the condition present in Equation 4.3.

$$\mathcal{F} = \{i : \mathcal{H}(i-1) < \mathcal{H}(i) > \mathcal{H}(i+1)\}, \text{ where } 0 \leq i \leq b \tag{4.3}$$

Figure 4.4 shows the histogram relative to the capture presented in Figure 4.2 *a)*. Image *a)* shows the histogram before Equation 4.2 has been applied, and *b)* after it has been applied; in both, the outlined circles represent local maximums obtained through Equation 4.3. As can be seen in *a)*, slightly after the first peek, the pixel count for some bins radically drops to 0 in a periodic fashion generating unwanted local maximums. However, after the histogram has been smoothed, some of the unwanted local maximums disappear while the main shape of the graphic is maintained.



a) **b)**

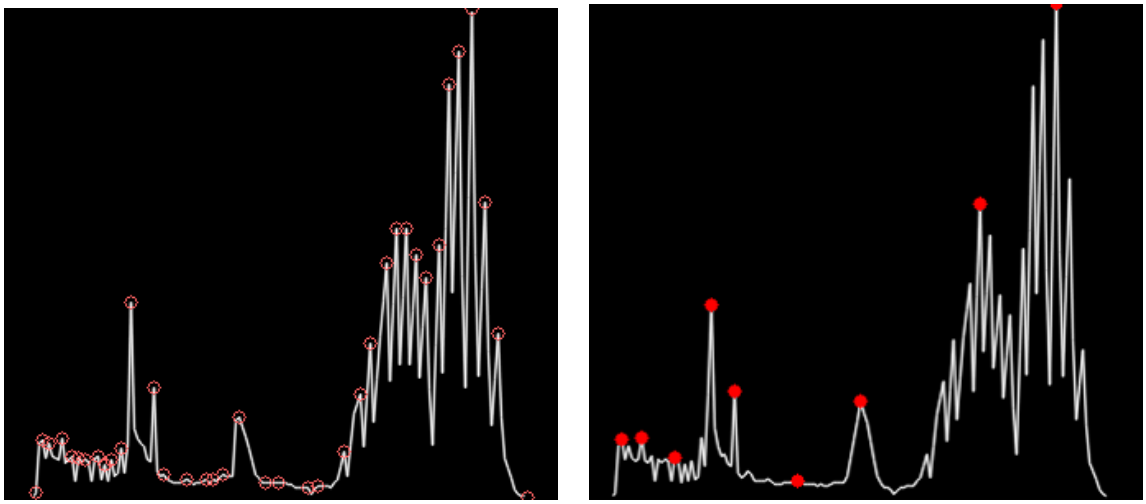
Figure 4.4: Histogram of a depth image, in a) before normalization, and b) after normalization.

For each detected local maximum, a range is going to be calculated with a starting bin and ending bin, which will be used to create slices of the environment that encompass all the bins in between, and segment the image as explained in the following sections. This process is associated with a certain computational cost, which increases linearly with the number of local maximums, therefore these should be reduced, maintaining only the most important.

In order to obtain only the most important local maximums, Equation 4.4 is applied to the previously obtained maximums in $\mathcal{F}$, discarding those that do not respect that condition, and storing only second order local maximums in $\mathcal{P}$. The second part of the condition was added because consecutive bins that present very similar counts have a tendency to generate unwanted second order maximums. Thus, in order to avoid this, for a bin to be a second order maximum, it has to have a count difference higher than ω pixels relative to its neighbours. The value used for ω is 200, which was obtained through testing and observation of different captures.

$$\begin{aligned}\mathcal{P} \quad = \quad & \{m_i : \mathcal{H}(m_{i-1}) < \mathcal{H}(m_i) > \mathcal{H}(m_{i+1}) \wedge \\ & (\mathcal{H}(m_i) - \mathcal{H}(m_{i-1}) > \omega \vee \mathcal{H}(m_i) - \mathcal{H}(m_{i+1}) > \omega)\} \\ & \text{, where } m_j = \mathcal{F}(j)\end{aligned} \tag{4.4}$$

The result of both sets can be seen in Figure 4.5. Image *a)* shows bins whose indexes were obtained using set $\mathcal{F}$, as outlined circles, and image *b)* presents only the ones considered more prominent, obtained by $\mathcal{P}$, as filled circles. As intended the number of levels proposed for expansion, for this particular example, was greatly reduced. The histograms presented in both images are related to the same capture, however they present slight differences due to the instants the captures were made.



a) **b)**

Figure 4.5: Result from the filtering of local maximums. In a) first order local maximums marked with outlined circles, b) Second order local maximums marked with filled circles.

Depending on the entropy of the environment and on the computational power available, the total number of maximums can be further reduced by applying other heuristics (e.g. allow only bins who present a count above a certain value). However reducing the number of local maximums selected for expansion, increases the probability of missing a peak where part of a person might be located and therefore, for this system, no more heuristic are applied after Equation 4.4.

4.2.2 Obtain Image Slices

After the most important levels of the histogram have been determined they have to be expanded in order to create depth slices of the environment, because humans and most objects have a certain thickness and do not occupy just one level of intensity in the image. In order to create these slices of the environment, the histogram is first analysed in order to obtain slices of the depth image's histogram, which are delimited by a starting bin and an ending bin. Then, a threshold is applied to the depth image, according to these values.

If we observe the histogram there are clearly regions that stand out, namely the peeks that mean that there is one or more objects occupying that depth in the image. Therefore

in order to properly segment these objects the slicing of the histogram should encompass the whole peek, from base to base, and preferably individual peeks should be included in different slices.

To do this an algorithm was developed where both the start and end bins are equalled to the second order maximums (peeks of the mounds) previously obtained, and are then expanded back and forth respectively until the bases of the mound or the limits of the histogram are reached. The bases of the mound are characterized by the changing of the growth direction, meaning that, starting from the local maximum, consecutive bins should present a decreasing behaviour (hill sides) and when this behaviour changes to increasing the base of the mound has been reached. Because the histogram presents so much irregularities, using all the bins of the histogram without any filtering would cause the algorithm to hit a wrong base, for example when there is a rapid decrease followed by a rapid increase in bin count ou vice-versa. Therefore, the values tested to determine if the base has been reached are only first order maximums, selected by Equation 4.3.

This process is repeated for each second order maximum previously obtained, and when the algorithm converges there will be as much slices as second order maximums. It is preferable to overextend these slices than to create slices that do not encompass the hole mound, because these are going to be converted into image masks as explained in section 4.3, and incomplete slices will generate masks with missing pixels.

Figure 4.6 uses a portion of the histogram presented in Figure 4.5 to demonstrate the process behind histogram slicing, where the outlined red circles represent the first order local maximums, the filled red circles represent the second order local maximums, marking the most prominent peeks which are used as the starting point for the slice expansion, and the inflection points mark the bases of each mound. The resulting slices are defined by a start and an end bin.

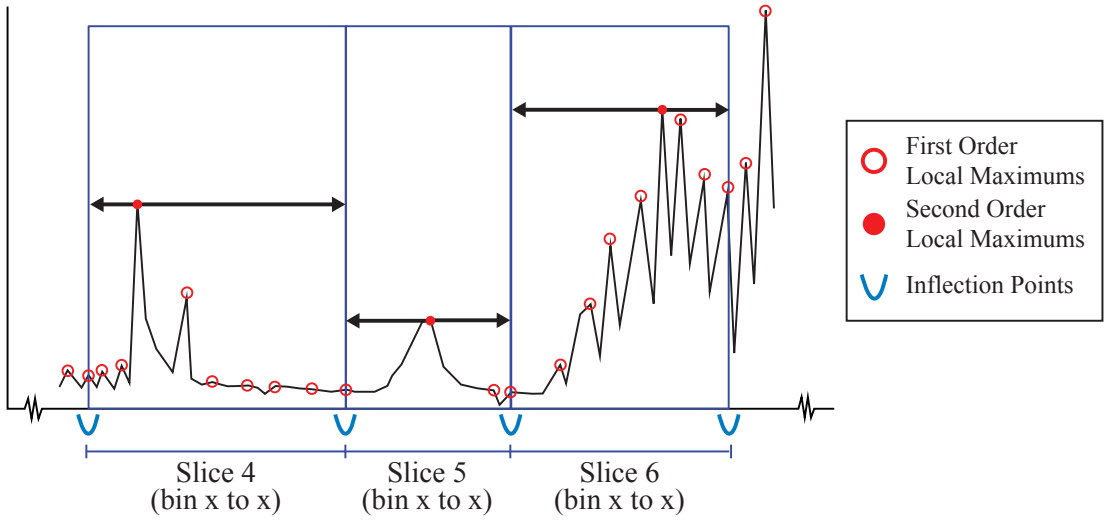


Figure 4.6: Diagram demonstrating the slicing process of the depth image's histogram.

To reduce the computational cost of the following stages another heuristic is applied to possibly reduce the number of obtained slices. By counting the total number of pixels present in each of the bins belonging to a given slice and summing them, it is possible to create a

minimum acceptable occupancy, in which slices that present a pixel count bellow this level are discarded. The threshold used in the proposed system is 400, which was obtained by observing several captures and manually identifying the slices that only retrieved random pixels caused by the lack of precision of the sensor, or very small objects.

4.3 Image Segmentation

Histograms are an useful representation of information contained in an image. They store a count for the number of pixels for a given bin, but do not include information relative to the pixels position. In order to convert the previously obtained slices to some format in which image-processing algorithms can be applied it is necessary to perform a thresholding of the depth image.

To perform the slicing of the environment, the slices obtained from the histogram are going to be used to create several masks. This is done by applying a threshold to the depth image, where the value of a given pixel, in the output of the threshold, is 0 if the depth image pixel's value is outside of the range defined by the beginning and end levels of a given slice, or 1 if the pixel's value is inside of that range. Equation 4.5 summarizes this process, where $\mathcal{M}_i$ represents the mask array for slice i, $\mathcal{I}_c$ is the converted depth image and $\mathcal{S}_{i,j}$ is the j^{th} value of the i^{th} histogram slice.

$$\mathcal{M}_i(u,v) = \begin{cases} 1 & \text{, if } \mathcal{S}_{i,1} < \mathcal{I}_c(u,v) < \mathcal{S}_{i,j} \\ 0 & \text{, otherwise} \end{cases} \tag{4.5}$$

The result of this conversion can be seen in Figure 4.7 in the form of the several masks obtained for each slice of the histogram. As can be seen these masks do not present correctly segmented individual objects, humans or not, therefore a second stage of image-processing is needed in order to retrieve the final ROIs for individual objects.

4.3.1 Segmentation through flood fill

The slices of the environment presented in Figure 4.7, cannot be used to properly segment objects in the image due to two problems: in cluttered environments, different objects that are present at the same depth will be encompassed in the same slice, generating incorrectly segmented objects; also, although there is only one object present in masks 4 and 5 of Figure 4.7, it is incorrectly segmented, since part of the floor and the person are considered the same object.

This stage of the algorithm is also used to recover from errors, such as when an object is divided between two or more slices, and to separate different objects present in the same slice. Using a flood fill algorithm, also employed by [9], applied to the depth image, entire objects can be retrieved, as long as they present significant distances to other objects.

In order to perform proper classification in the following stages, each ROI retrieved should be loyal to the real shape of the object and should only mask individual objects. To solve this the following process is applied for each slice: using the Suzuki, S. algorithm [33], the contours of isolated regions in the binary image are retrieved, and, for each one, its centroid is used as a seed for the flood fill algorithm. The resulting regions will ultimately be the regions of interest, *ROIs*, used throughout the rest of the project pipeline for classification and tracking.

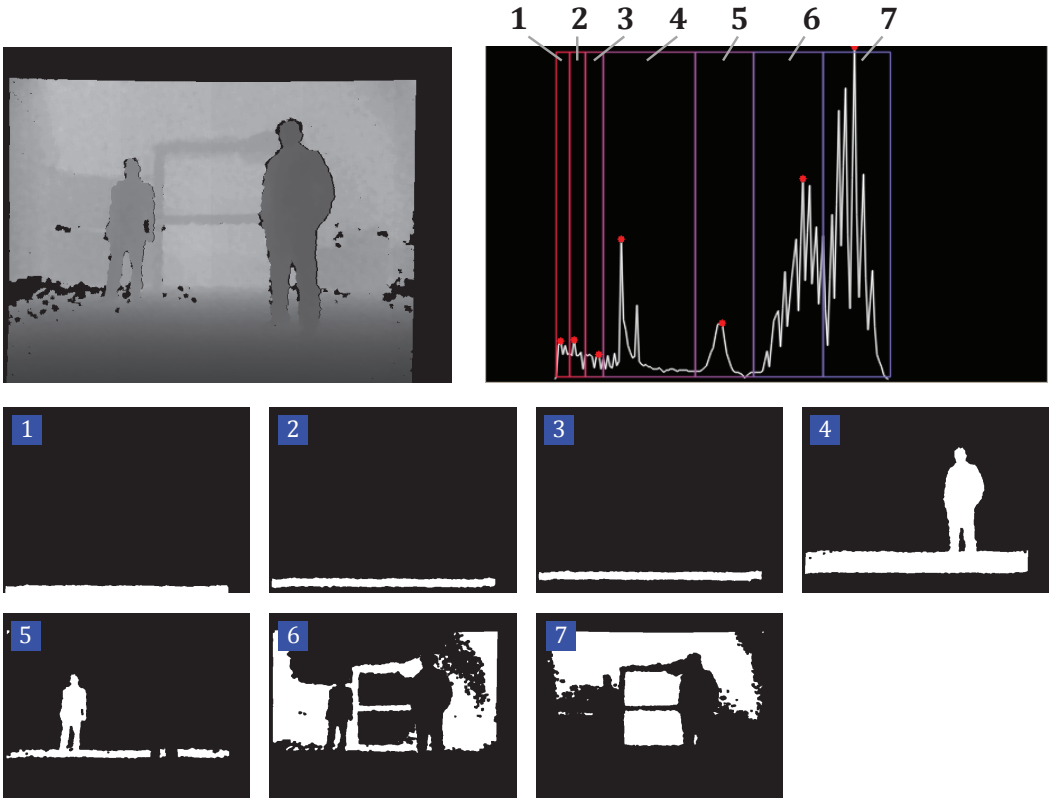


Figure 4.7: Result from the slicing process in the form of masks.

It is possible to observe in Figure 4.7, *slice 4* for example, that the centroid of that mask would not be present over the person, due to the presence of the floor in the slice. To solve this, the obtained slices are cut through a process explained in subsection 4.3.2, which erases the floor leaving only the person masked and enabling the retrieval of a correct seed.

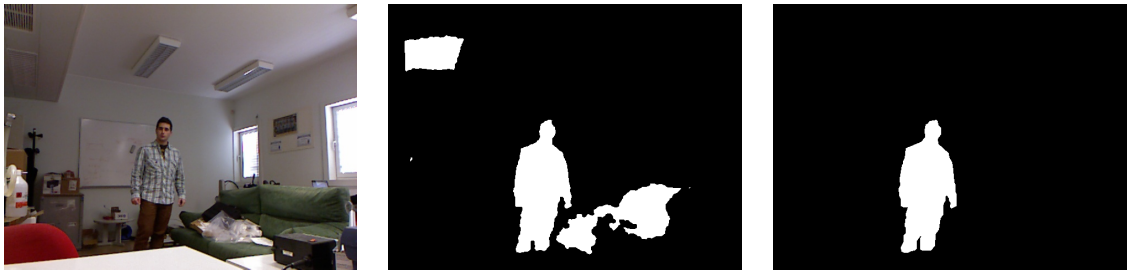
Flood fill algorithms are common among image-processing techniques, because they are able to extract contours of objects in images by expanding a small selection of pixels until the edges of the object are reached. The algorithm starts by adding the seed pixel to a list, and, then, every neighbour of this pixel is tested, according to a given heuristic, to determine if it is part of the same component or not; if it is, it is added to the list. The algorithm performs this same test for every pixel added to the list, until there are no more unclassified neighbour pixels. This enables the extraction of contours of objects in an image.

The heuristics are used to determine if a neighbour pixel is connected to the observed pixel known to belong to the component and this is done by determining the closeness for the value of the two pixels and deciding upon this value. The first heuristic tested uses the closeness of the value between the currently observed pixel and neighbour pixels. The second heuristic uses the closeness of the value between the seed pixel and the neighbour pixel.

Let $\mathcal{I}_c$ be the converted depth image, s the seed pixel, o a observed pixel, n a neighbour pixel, η the lower limit for the closeness between the values of the tested pixels, and ρ the higher limit. Equation 4.6 and Equation 4.7 represent respectively the first and second heuristics, as detailed in [34].

$$\mathcal{I}_c(o_u, o_v) - \eta \leq \mathcal{I}_c(n_u, n_v) \leq \mathcal{I}_c(o_u, o_v) + \rho \tag{4.6}$$

$$\mathcal{I}_c(s_u, s_v) - \eta \leq \mathcal{I}_c(n_u, n_v) \leq \mathcal{I}_c(s_u, s_v) + \rho \tag{4.7}$$



a) **b)** **c)**

Figure 4.8: Example of a correct ROI: in a) color image, in b) slice of the environment and in c) resulting ROI generated through flood fill.

In Figure 4.8 there is an example of a color image capture in *a)*, one of the resulting slices in *b)* and one of the ROIs retrieved using the flood fill algorithm in *c)*. Is is possible to see that, although the image slice encompassed different objects (such as the couch and part of the ceiling), the presented ROI masks the person correctly and individually.

The example shown in Figure 4.8 was taken in a specific situation where the person is not in direct contact with any object. However in captures where two objects are very close, such as a person's feet and the floor, a side-effect of using a flood fill algorithm can occur, as can be seen in Figure 4.9. It is often referred to as *overflow*.

a) **b)**

Figure 4.9: Example of an incorrect ROI: in a) depth image, in b) slice of the environment and in c) resulting ROI generated through flood fill.

Overflows can occur when the closeness of two pixels values is enough for the flood fill algorithm to connect them in the same component when they should not be connected, which leads to other pixels being connected until the algorithm converges. Because the value of a pixel is related to its depth, overflows usually occur when two different objects are close or even in contact.

To counteract this effect as much as possible, the second heuristic was chosen (Equation 4.7), where a seed pixel is used to determine the closeness of two pixels. This heuristic allows for a more controlled flooding because one the control values is fixed, defined by the seed, while in the first heuristic (Equation 4.6) the value from both the observed pixel and the neighbour pixel change, increasing the probability of overflow. Furthermore, in cases where overflows cannot be avoided, using the second heuristic ensures that this event has less of a deteriorating effect. If, for example, the other heuristic had been applied to the depth image in Figure 4.9 the entire floor would have been flooded instead of just part of it.

When the flood fill algorithm converges it returns a vector of connected components, also described here as ROIs, and one slice might have more than one component. These ROIs are then put through a filter where those whose area is smaller than a given control value are discarded, leaving only individually recovered objects that have a significant size, while discarding regions associated with small objects in the environment.

As stated before, it is important for the final ROI to match the real shape of the object it is masking. In cases such as the one present in Figure 4.9, it is clear that the ROI in *c)* does not match the real shape of the person. To solve this kind of problems, a solution is presented in the next section.

4.3.2 Detecting Human Limits

Detecting the limits of an object is a common problem among object detection systems. For example, if a person is standing up where do his/her feet end and do the floor start? Or, if the person is holding an object, what part is the object and what part is the person.

This problem requires specific solutions fitted to overall system.The presented solution, using a flood fill algorithm to determine the final shape of the object, is inspired by Xia, L., et al. [9] work. As us, the researcher faced the problem described in the previous section as overflow.

To resolve overflows Xia, L., et al. employs a filter to extract the vertical boundaries in the image, with focus on the person's feet and the ground, and uses it to enhance these boundaries and avoid overflows. By traversing the filter defined in Equation 4.8, the author is capable of detecting vertical boundaries present in the image. The result of this filter is then thresholded in order to keep just the strongest boundaries and added to the original depth image, creating a depth image with significant brightness differences on vertical boundaries. This process enables the use of a flood fill algorithm, with less probability of occurrence of overflows.

$$[1, 1, 1, -1, -1, -1]^T \tag{4.8}$$

Xia, L., et al. perform several image preprocessing stages in his proposed method, namely nearest neighbour interpolation in order to fill discarded pixels (pixels with brightness 0) and a 4×4 median filter to smooth these artificially filled pixels and reduce noise. Furthermore the researcher works with the full precision depth array (16 bits). In the proposed work, due to the reasons exposed in section 4.1, the preprocessing done over the original depth image is quite different and therefore the result of the filter proposed by Xia, L., et al. does not produce results as useful as in the researchers system.

Figure 4.10 presents the results obtained by the same filter, where *a)* and *b)* show images taken from Xia, L., et al. paper [9] in unfiltered and filtered versions respectively, and *c)* and *d)* show an unfiltered capture from our environment and the result obtained by the filter.

The images presented by the researcher are not very explicit, but it is possible to see that, after the filter has been applied, some brighter horizontal lines appear in image *b)*.

Several tests were performed, with different filter sizes and different threshold values. It is possible to see in Figure 4.10 *d)* that the filter indeed detects vertical boundaries, but it did not efficiently detect boundaries on the planar intersection of the feet and the floor. The areas where the filter shows the existence of considerable vertical boundaries are already areas where the flood fill algorithm responds correctly and does not create an overflow. Therefore, the use of this filter does not bring any advantage to the system.

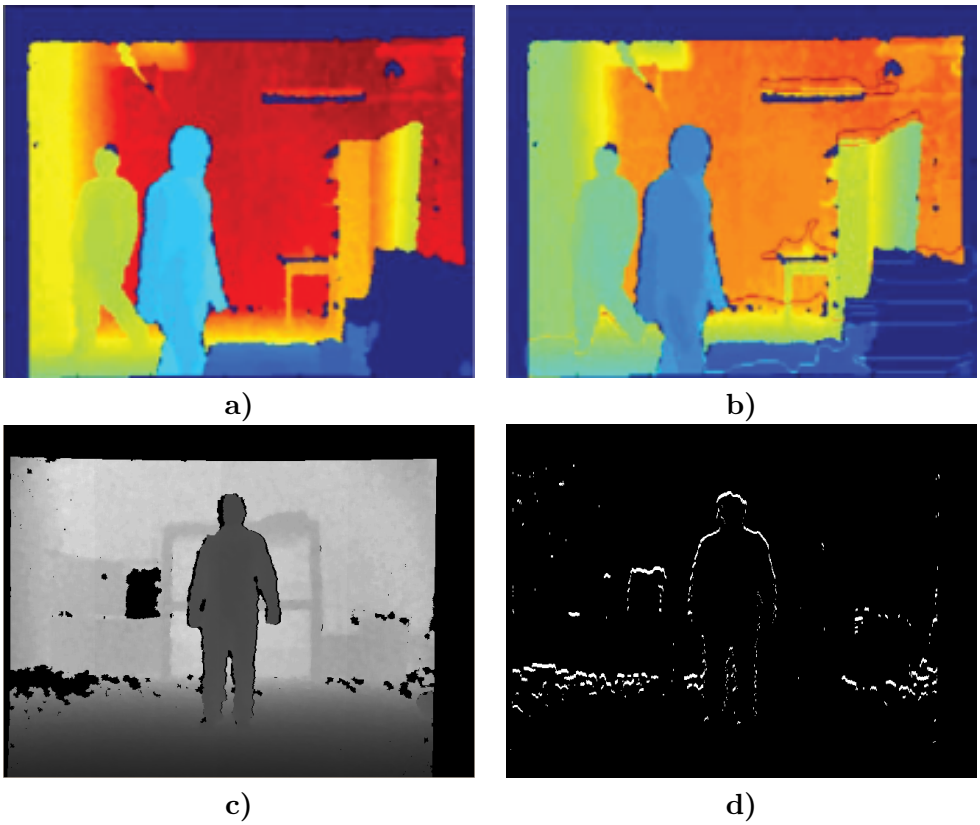


Figure 4.10: Depth images before and after the application of the vertical transition filter: images a) and b) present changes using Xia, L. et al. filter [9] before and after respectively, image c) a capture from our lab using no filter and image d) the result obtained by the filter.

Nevertheless, the problem of the overflow is still present so another solution was designed. Due to the nature of this project, which is supposed to be implemented in the CAMBADA@Home service robot, some assumptions can be made, for example the height of the camera and its tilt will be know through communication with the robots components such as the *pan & tilt* support for the cameras and the adjustable height. However, because the physical platform itself is still under development, the cameras support is not functional yet, and the height of the robot is static. Therefore the value for the height of the camera was obtained manually by measuring the height from the floor, to the depth sensor of the Kinect, and the camera is assumed to be parallel to the floor at all times.

The ROS middleware provides some image geometry support functions that are useful in

such situations. For example, given a target height difference Δr, in real world coordinates relative to the Kinect, and the desired depth value d, it is possible to calculate the row r in the image that corresponds to that height. This enables the erasing of pixels from the ROIs mask, bellow this row.

$$r = \frac{d \times \Delta r}{f_y} \tag{4.9}$$

Equation 4.9 presents the equation used by ROS to determine this row r. As mentioned before the height of the Kinect is known, consequently Δr can be obtained by subtracting the desired height of the cut, c, by the height of the Kinect, h. This height difference is then multiplied by the depth d, and the result is divided the focal length of the Kinect lens in the vertical axis, represented by f_y. Although the depth array has been converted to 8-bits, it is still possible to obtain the depth of a pixel by solving Equation 4.1 in order to $\mathcal{I}_o$ and because the depth must be in Cartesian coordinates, the final value of d is equal to the original depth value in millimetres converted to meters, as shown in Equation 4.10.

$$d = \mathcal{I}_o \times 0.001 = \left(\psi \times 1000 + \frac{\mathcal{I}_c(u,v) \times (\gamma - \psi) \times 1000}{b}\right) \times 0.001 \tag{4.10}$$

Figure 4.11 illustrate this procedure. On the left side one can see a diagram with every measure required, properly identified, and on the right side where the cut is performed on the mask.

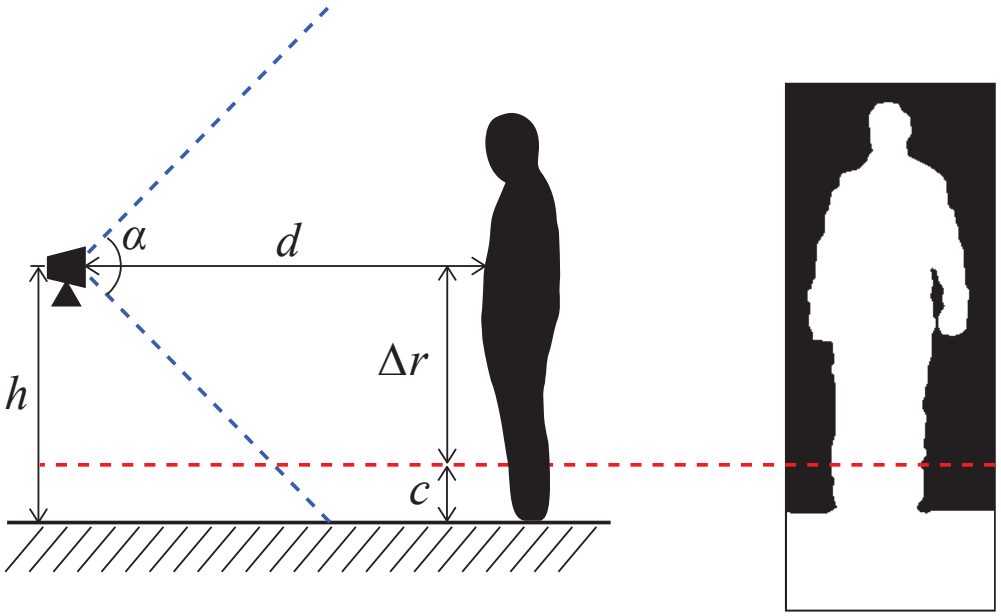


Figure 4.11: Diagram explaining the cutting process of the ROI.

Cutting the mask just above the floor level is not enough, it has to be done slightly above this plane due to the brightness differences allowed by the flood fill algorithm. For example it is better to perform this cut at least 0.3 meters above the ground plane.

This solution solves the problem of overflows from the persons feet to the floor, however it does not solve overflows in cases where the person is touching or holding some object with

the hands, for example.

4.4 Regions of Interest Filtering

Until this point in the processing pipeline no restrictions specific to the human shape have been applied, in order to ensure that no ROIs are not discarded prematurely. Now that well defined regions are available some restrictions related to human shape can be enforced. In [4] some constrains are applied during the segmentation process, however, since in the proposed method there is a separation between segmentation and template-matching for classification, the filters described in this sections cannot be employed earlier in the pipeline.

Since one of the main objectives of this system is the human detection and tracking in unconstrained environments, the proposed filters were chosen carefully and do not assume that the person is in a specific stance or position in the capture. Given the right conditions, mainly limited by proper segmentation of the human form without overflows, it is possible for a detection to occur with persons standing up, sitting down on the floor or chair, or even partially occluded.

The restrictions proposed ahead are not meant to be thorough because they only collect information from the shapes and sizes of the ROI and not from the available images. Therefore it is preferable for this stage to allow more non-human regions to pass through than to discard human regions prematurely.

4.4.1 Proportion Filter

The first stage of the rejection process discards ROIs based on their proportions. Because the size of an object changes proportionally along the depth axis due to the perspective effect, this filter is depth-invariant.

There is a know relation between a person's height and its arm span, in which a children arm span is $1cm$ shorter than its height, an adolescent arm span is equal to its height and an adult arm span exceeds its height by more than $5cm$.

These are approximate measures that vary according to the persons physical shape, however they describe a trend that is useful for this filtering stage. The proportion of a ROI can be calculated by dividing the ROI's bounding box width by its height. Assuming the contour of the person is correctly extracted (no overflows), if the persons has its arms down close to the torso or is standing sideways, the width of the ROI will be smaller than its height (resulting in a value lower than 1 for the proportion), and if the persons has both arms completely extended to the sides, because the previous process cuts the ROI above the floor plane, the width of the ROI will be bigger than the height (resulting in a value slightly bigger than 1).

The situations described before present normal situations with no occlusions, however the system has to be able to cope with occlusions and therefore the minimum and maximum values for the proportion must be adjust up to a certain limit for extreme situations.

The highest value for the proportion is achieved when a person has its arms wide open and is partially occluded starting from the bottom. For these cases the ROI will be incomplete and the real height of the person cannot be determined. The minimum value occurs when the persons is sideways, possibly occluded from the sides, resulting in a thinner ROI than the actual persons width.

The occlusions considered for these extreme cases do not include cases where the head and shoulder of the person are not visible, because without direct vision from these features

the classification cannot be performed, as explained in chapter 5.

Figure 4.12 shows examples of the poses described before, where images from *d)* to *f)* present the ROIs without occlusions, and images from *j)* to *l)* present ROIs with occlusion. All these are accompanied by the respective color captures to show the object causing the occlusion.

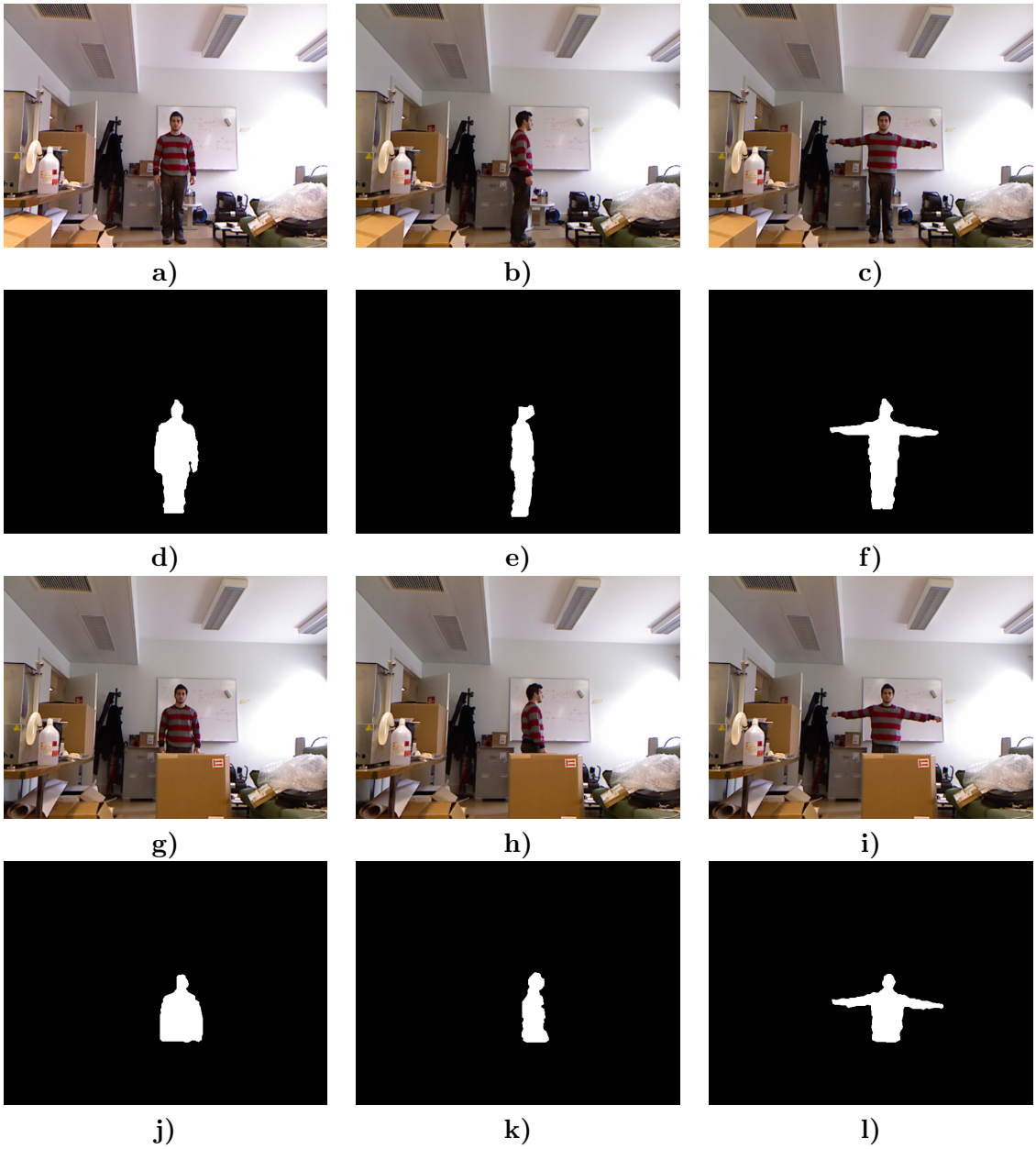


Figure 4.12: Example of possible poses with and without occlusion, and the respective ROIs.

For each of the poses displayed in Figure 4.12, the bounding box surrounding the ROI for four different individuals was measured in pixels and the average result for each case is presented in Table 4.1. From this table it is possible to confirm which cases generate the

highest and lowest values for the proportion. Because this rejection stage is not supposed to be too rigid, the values used for the minimum and maximum acceptable proportions are an approximation from the obtained measures but less restrictive, such as a *minimum proportion value* of 0.1 and a *maximum proportion value* of 1.7.

Region of Interest	Proportion ($\frac{Width}{Height}$)
front (no occlusion)	0.4008
sideways (no occlusion)	0.2321
arms open (no occlusion)	1.0028
front (with occlusion)	0.6078
sideways (with occlusion)	0.3900
arms open (with occlusion)	1.5547

Table 4.1: Proportion values for each detection presented in Figure 4.12.

4.4.2 Area Filter

The second filter rejects ROIs based on their occupied area, which is very helpful to eliminate large detected ROIs such as walls that, despite their human-like proportions, occupy a very large area.

Because we are dealing with images, and not point clouds, the perspective effect causes objects close to the camera to have a higher occupied area in the image than the same objects far away from the camera. Therefore, unlike the first rejection stage, this stage is not depth-invariant.

To have a notion of the area a person occupies in the image, at different depths, an experiment was created in a long hallway, which was marked meter by meter starting from the Kinect and up to 7 meters. Then several subjects were asked to stand at each of the marks and measures were taken creating the graphic present in Figure 4.13, where the xx axis represents the distance to the camera and the yy axis the area measured. The ROI segmentation from the fifth meter forward presented lots of irregularities therefore measures where taken until the subject was undetectable, resulting in several measures between the fifth and sixth meter.

Using the obtained data, a trend line, obtained by Equation 4.11, was calculated.

$$a = 96882 \times e^{(-0{,}561 \times d)} + \rho \tag{4.11}$$

In this equation, a is the maximum area a human object should occupy at depth d. Because the number of individuals used to create this trend line was not very large, a control value ρ is added to the final value to perform a finer tuning. When overflows occur because the person is near or touching another object, the area of the ROI may change considerably, and thus it was decided it is better not to discard these ROIs just because their contour is not completely true to the real person's shape, for example in cases were a slight overflow occurs.

4.4.3 Other Filters

Another pair of filters, not related to human characteristics, were implemented in order to further discard regions and enhance performance. In the process presented in subsection 4.2.2,

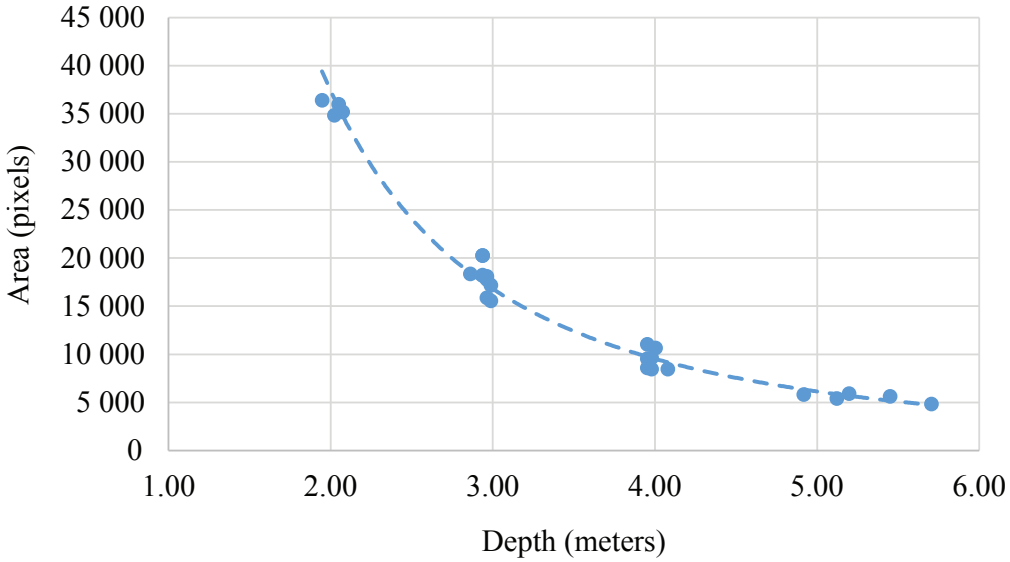


Figure 4.13: Measured area for human ROIs, and resulting trend line.

it is possible for parts of the same object to be segmented in different slices if it is present throughout several depth levels in the image, thus resulting in several seeds for the same object. When these seeds are used in the flood fill algorithm, if they are very close, there is a high probability that the resulting flooded ROIs mask same object, resulting in overlapped regions. An example of this type of situation can be observed in Figure 4.14, where *a)* presents the depth image's histogram and it can be seen that the forth and fifth slices share a small peek, which generates masks *b)* and *c)* with different parts of the same person present in both (person on the right), resulting in the ROIs shown in *d)* as outlines, and red circles marking the center of each ROI.

To filter overlapped regions two more rules are enforced before finalizing the ROI retrieval process. ROIs whose weighted center is closer than a given search radius are discarded, as well as ROIs whose overlapping area is bigger than a given area.

To perform this, all ROIs are compared with each other, and in cases where their center is near or they present a considerable overlapped area, the region with the smallest area is discarded. Comparing ROIs using these heuristics effectively reduces the total number of overlapped ROIs on cluttered captures, however, it also discards smaller ROIs that might have been correct.

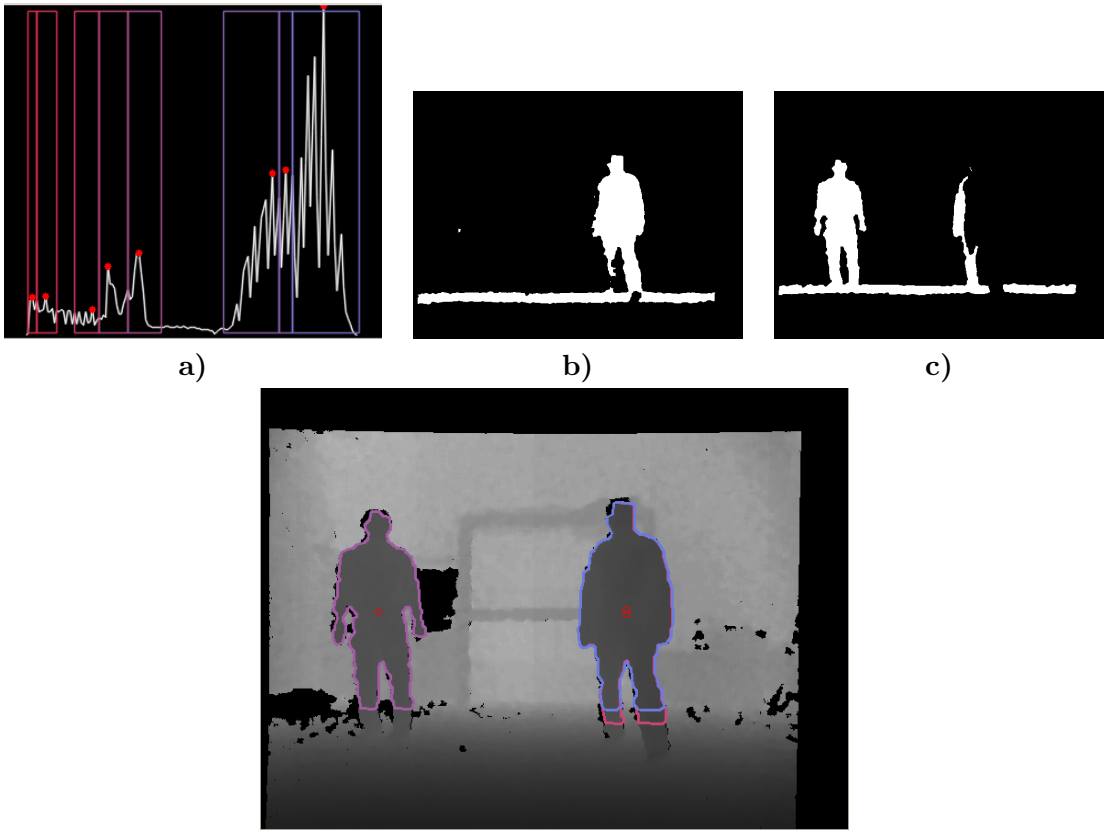


Figure 4.14: Example of overlapping ROI due to incorrect histogram slicing. a) Depth image's histogram, b) and c) mask derived from the forth and fifth slices of the histogram, and d) final ROIs represented in the depth image.

4.4.4 Result of the Proposed Filters

The effects of each filter can be seen in Figure 4.15 where the capture of a cluttered environment is shown. The color image is shown in *a)* and depth image in *b)*, the unfiltered ROIs are shown as outlines in the depth image in *c)*, followed by filtered versions: in *d)* by proportion (see subsection 4.4.1), in *e)* by the previous filter plus area filter (see subsection 4.4.2) and in *f)* by the previous filters and other filters (see subsection 4.4.3).

As can be seen in the passage from image *c)* to *d*, the proportion filter eliminates the largest non-human ROIs, mainly present in walls or big objects where the flood fill algorithm grew until the borders of the image were reached. From image *d)* to *e)*, using the area filter, the region on the left side of the image, that also overlapped with the ROI of the person in the center, has been discarded due to its large occupied area at a high distance from the camera. Between the last two images, *e)* to *f)*, the ROIs that presented a considerable overlap, caused by overflows in objects that are too close together or even touching, such as the ones present on the table in the left side of the image, were discarded.

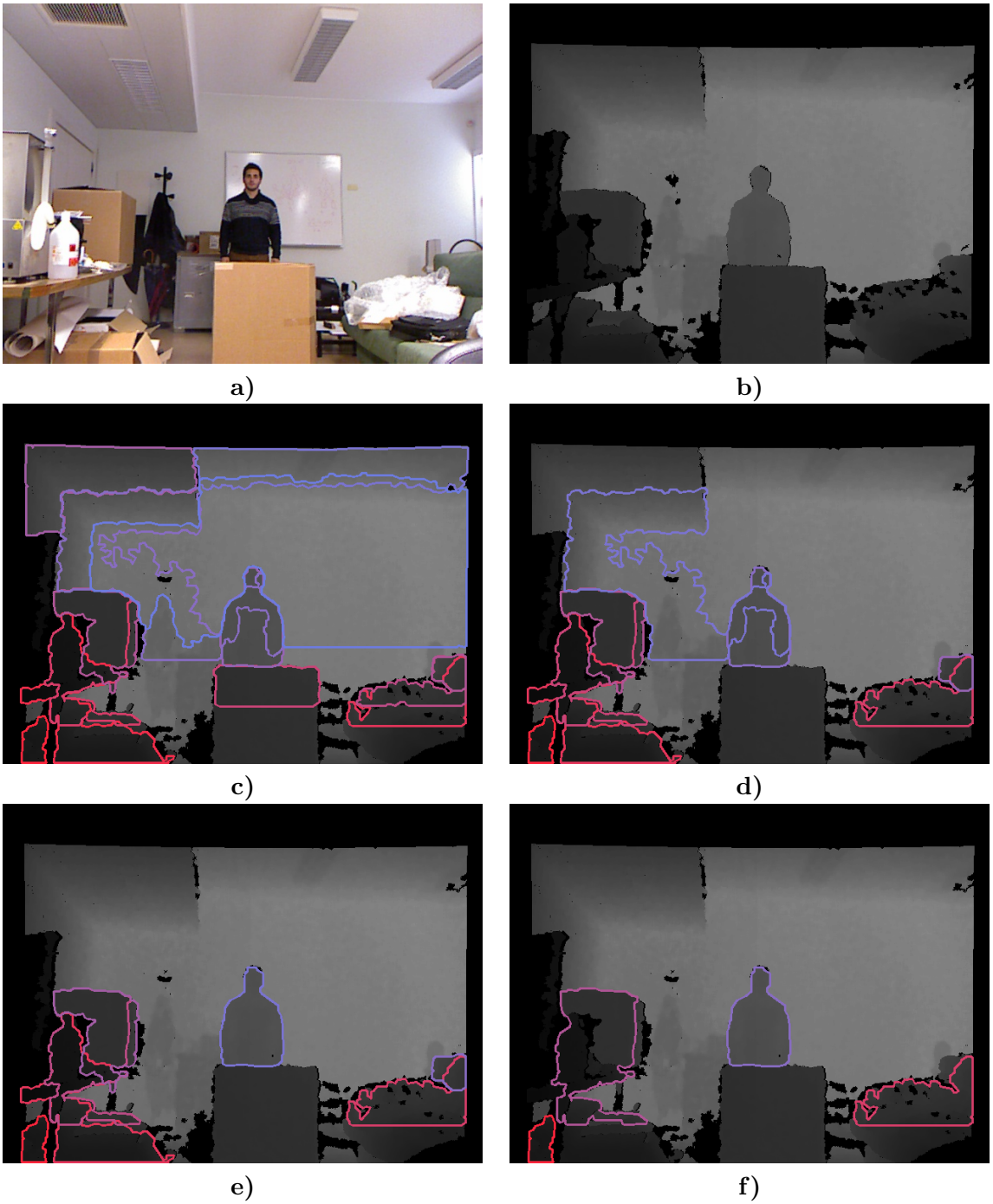


Figure 4.15: Results of the different filters described in this section. *a)* color image capture, *b)* depth image capture, *c)* unfiltered ROIs outlined, *d)* ROIs filtered by proportion, *e)* ROIs filtered by proportion and area, and *f)* ROIs filtered by proportion, area, and other filters.

Chapter 5

Classifying Regions of Interest

The majority of techniques used to perform people detection in images can be divided into two approaches, either the system searches directly for human features in the image (ex. skin color, eyes) ([17], [18], [8], [9], [6]), or it uses techniques, such as, background extraction or template matching to separate objects between foreground and background, making afterwards their classification as human or non-human ([11], [16]).

The method proposed in this work belongs to the second approach, since objects are segmented from the background through the analysis of the histogram of the depth image, as described in chapter 4. The result is a list of ROIs, also described as human-candidates. The second part of this method is related to the classification of the ROIs, as human or not-human, using a template matching technique, inspired by [16], [9], [4].

The classification is performed using a template matching algorithm used by the Robocup Middle Size League team, Brainstormers Tribots, also used by the CAMBADA MSL team in the same league, as a form of self-localization for the robots. The Perfect Match algorithm [35] is a fast and effective localization algorithm which was adapted to this system as a template matching algorithm in a novel way.

A diagram is presented in Figure 5.1, where the three main stages of the classification process can be seen, integrated in the Image Analysis node. The node consumes the information publish by previous nodes, namely the processed color and depth images, as well as the ROIs previously obtained. It then creates the Distance Transforms required to perform multiple template matching and concludes with region classification. The ROIs classified as human are then passed to the next stage, People Tracking.

5.1 Template Creation

In the early stages of development of the proposed system, the template used to perform the matching was a copy of Xia. L. et al. template used in [9], because, as the author states, the area of the head and shoulders is the less deformable part of the body. As the system entered its more mature stages the classification algorithm, presented in section 5.2, required more accurate data and so the pixelated version of Xia's template did not provide results precise enough.

One important fact when working with sensors it is not to assume that their output is predictable. A good example of this was seen in this project while using the Kinect camera: as explained before, the depth image shows solid objects in the image, however, there are

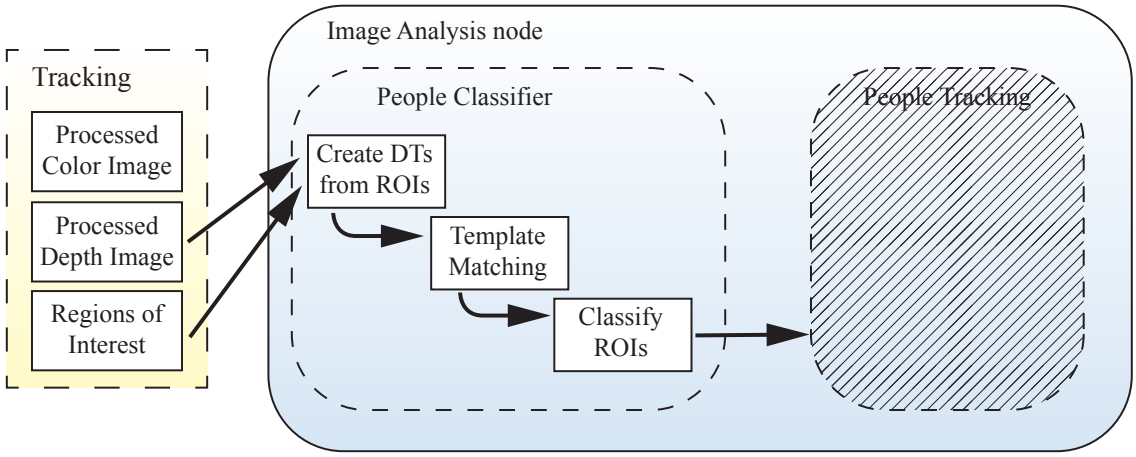


Figure 5.1: Diagram detailing the classification process of ROIs.

some materials that the Kinect is incapable of reading. While the incapacity of the Kinect to detect windows and some metals, due to reflection, is not a limitation for this project, its inability to read depth values from the person's hair is an important factor when performing classification using the head shape, particularly when the classification is performed for persons facing sideways or back, where the hair occupies most of the persons head. Furthermore, the physiognomy of the persons is also an important factor, where for example Xia. L. et al. template did not match correctly on some of the subjects tested for this project, which might be caused by lens distortion or bad precision of the sensor.

These factors combined, plus the fact that the system requires different templates to estimate the pose of the person, led to the need to create specific templates. A dedicated template creation algorithm was developed using the contours obtained from the Kinect's depth image, ensuring that the templates are not what we as humans think the shape of a head is, but what the Kinect is actually capable of capturing.

The fact that the templates used in this system were obtained through measurements obtained using the sensor itself, creates some similarities with the approach taken by other works who use Learning Techniques to perform People Detection.

In order to generate a template that is an average of several subjects, a Distance Transform map (DT) is created from the contour of the head and shoulders of several test subjects. These maps are then summed, creating an average DT of several subjects. Let $\overline{\mathcal{M}}$ be the average DT, $\mathcal{M}$ the set of DT samples used. Equation 5.1 is used to create an average map, where each DT was previously resized to a size of 100×100 pixels, before being added.

$$\overline{\mathcal{M}} = \mathbb{E}[\mathcal{M}] \tag{5.1}$$

This average map cannot be used directly to perform matching with other maps. It must pass through some image-processing stages before it is transformed into a set of points usable by the algorithm. The average map is normalized, a threshold is used to retrieve only the relevant values and a thinning algorithm is applied in order to create a refined template consisting of a thin line which can be transformed into a set of points.

The DT can be interpreted as an occupation map, where the value of each pixel is equal

to the Euclidean distance of the closest pixel of value zero. In the proposed system, a pixel of value zero means it is in the contour of the ROI.

An average depth map was created for each posture (front facing, left facing, right facing and back facing), using samples from different depths since the precision of the Kinect lowers considerably with the increase in distance to the camera, creating very different contours for subjects standing close and far from the camera. Figure 5.2 shows some of the contours used to create the final templates: *a)* shows contours from persons facing forward, *b)* from persons facing right and *c)* from person's backs. It is important to notice the deteriorating effect of the hair and the distance to the camera in some of the samples, such as the second and fifth contours in *b)* or first, second and forth contours in *c)*.

To enable the visualization of the resulting average distance map (see Figure 5.3 *a)* to *c)*), the values are normalized to an 8-bit encoding, and a threshold is applied to the lower values, where the presence of contours is stronger (Figure 5.3 *d)* to *f)*), creating a thick outline that defines the shape of template. Increasing the values encompassed by the threshold will thicken the outline and reducing the number of values may cause the outline to contain gaps. As explained before one of the reasons for this procedure is the creation of templates more loyal to what the Kinect sensor captures, therefore a final transformation is applied to the image where a thinning algorithm refines this imperfect outline, creating the final template (see Figure 5.3 *g)* to *i)*).

It is important to note the effect of hair on the templates. In the template for sideways postures (see Figure 5.3 *h)*) the head is not very rounded on top, where most of the hair is located, and specially on the template for postures facing backwards (see Figure 5.3 *i)*), where most of the head is covered with hair, and also because a person's posture is not perfectly upright, normally leaning a little forward.

The templates obtained from the thinning process are not the final ones. It is possible to observe that this process creates some artefacts in the images, close to the edges and, as explained ahead, the shape of the templates heavily conditions the classification, so not the entire lines will be used in the final templates.

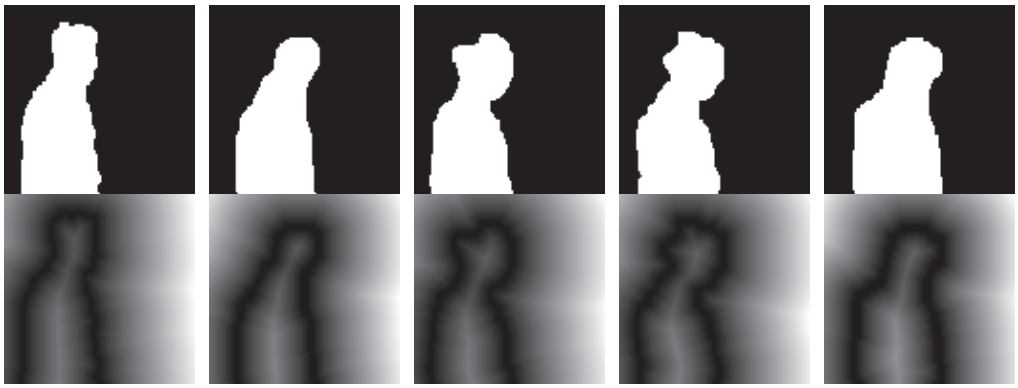
The final stage of the template creation process is performed in an image editing software, where the lines closer to the edge are erased and the template is centered and trimmed. The final templates are shown in Figure 5.4, where *a)* is the template for persons facing the camera, *b)* for persons facing right, *c)* is a reflection of *b)* for persons facing left, and *d)* the template for persons facing backwards to the camera.

Throughout the development of the system several templates were tested and an important conclusion was reached. Due to the effectiveness of the fitting in the template matching stage, it became relatively easy to find a point in the ROI where the template fits with low error, even in cases where the ROI is not human. Controlling the amount of shoulders covered by the template helps hinder the fitting of the template in ROIs that are not human. Because the Ω shape of Figure 5.4 *a)* is naturally harder to fit on templates due to its accentuated curves, the amount of shoulders covered is lower than the one covered by templates *b)* to *d)*, which are easier to fit in contours.

5.2 Template Matching

The classification problem, when dealing with human detection systems, presents several approaches that are mainly divided between template matching and machine learning

a)



b)

c)

Figure 5.2: Examples of head contours and respective maps. a) facing forward, b) facing right and c) facing backwards.

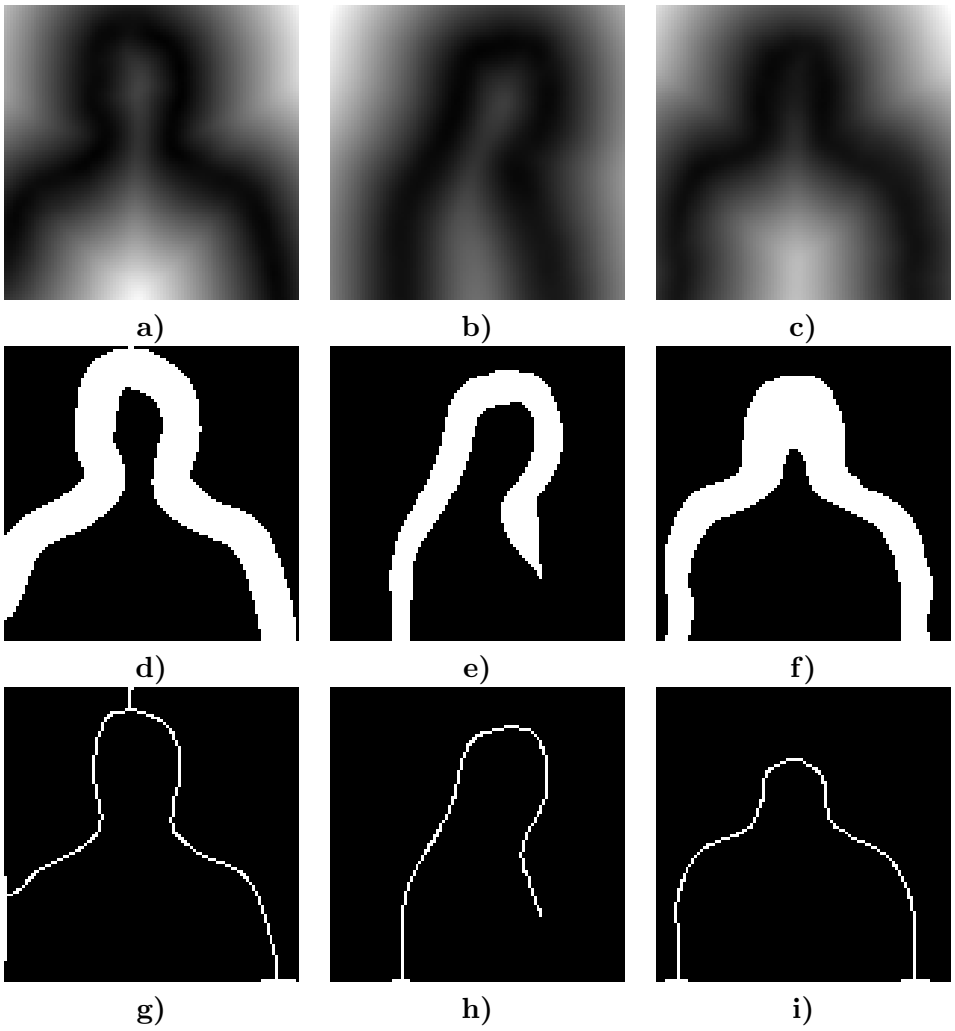


Figure 5.3: Stages of the template creation process.

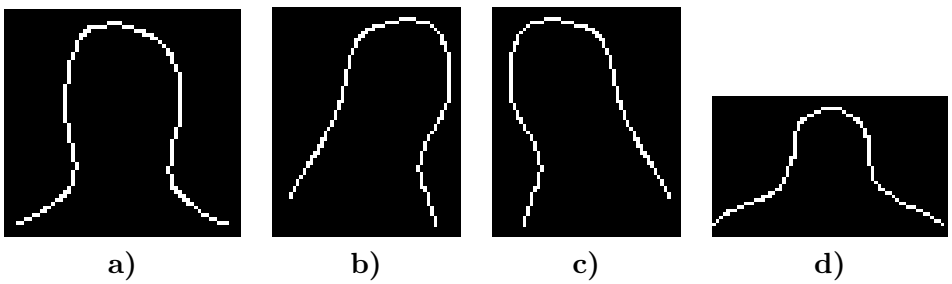


Figure 5.4: Final templates used in the classification stage.

techniques. Human classification through machine learning techniques is a popular approach among researchers, such as [17], [18], [20], [19], [6] or [8], were the main choices differ in the combination of features and learning methods used to perform the classification. This choice

is far from trivial, as can be conjectured from the overall number of approaches taken by researchers, the most popular having been presented in chapter 2.

The other approach, template matching, is also supported by some researchers, such as [11], [9] or [21], as well as the proposed method. However, the proposed solution also shares some traits with learning techniques, since the creation of the templates relies on information generated by the camera.

The choice for this type of approach was done considering the image analysis background of the researcher, and the fact that the time required to develop a learning method for classification could not be enough, due to the fact that these approaches usually require more time than others since the classifier has to be trained.

The classification process only draws information from one source, the depth and color. This section describes the template matching stage using the depth image, where the templates presented in section 5.1 are fitted over each ROI retrieved in the previous stage of the pipeline, covered in chapter 4. The process starts by resizing all the used templates, so that they respect human sizes according to the depth of the ROI. Next, the Perfect Match [35] algorithm is used to process the template over the image, in order to find the position that best fits and, therefore, generates the lowest matching error. The classification as human or non-human is performed by contemplating a classification score, which is obtained using the error value for the matching and its variance, as explained in section 5.3.

5.2.1 Template Resizing

When working with images from monocular cameras an important factor to take into account is the fact that the size of a given object in the image, whether it is a color, thermal or depth image, depends on its proximity to the camera, being this is known as the perspective effect. When performing template matching in images this is an important factor to take into consideration.

Previous human detection systems, from researchers such as [14], [16], [9] and [21], solve this problem by creating an *image pyramid*, where the base level is composed by the image with the original resolution, while lower resolution copies stack on top of this level, creating a pyramid-like vision of the scene. This allows for a single template to be matched on several images with different resolutions to reflect the perspective effect. The number of levels of the pyramid as well as the re-sampling method for the different resolution image should be adapted for each system.

On the proposed system this method was not adopted mainly for two reasons: first, the computational cost to perform matching on each level of the pyramid is higher than performing a single match, and second, even if the template fits on a certain level of the pyramid there are no guarantees that the object has the correct size due to the resizing that occurs for each level.

Instead of resizing the DT of the ROI we propose to resize the template itself, to counteract the perspective effect of monocular cameras, allowing for a single matching instead of multi-level matching. At he same time, restricting the size of the ROI by testing the template only for sizes possible to humans, reduces the number of misclassification. This approach is also supported by Guan, F. et. al. [4], where the researchers propose to search for human features taking into account the height, width and thickness of the human body. In the proposed system these constraints are loosely enforced in the filtering stage of ROIs (section 4.4), and enforced again in this stage by resizing the template to human proportions, depending on the

depth of the ROI.

Another advantage of performing resizing of the template instead of the image is that, since the template used is actually converted into points, changing its size does not generate loss of precision has happens when resizing images.

To perform the resizing of the templates an equation was obtained by manually fitting one template over the ROI and observing how the scale factor behaves. For the rest of the templates, the behaviour of the function should be the same, making it only necessary the addition of an initial value, specific to each template.

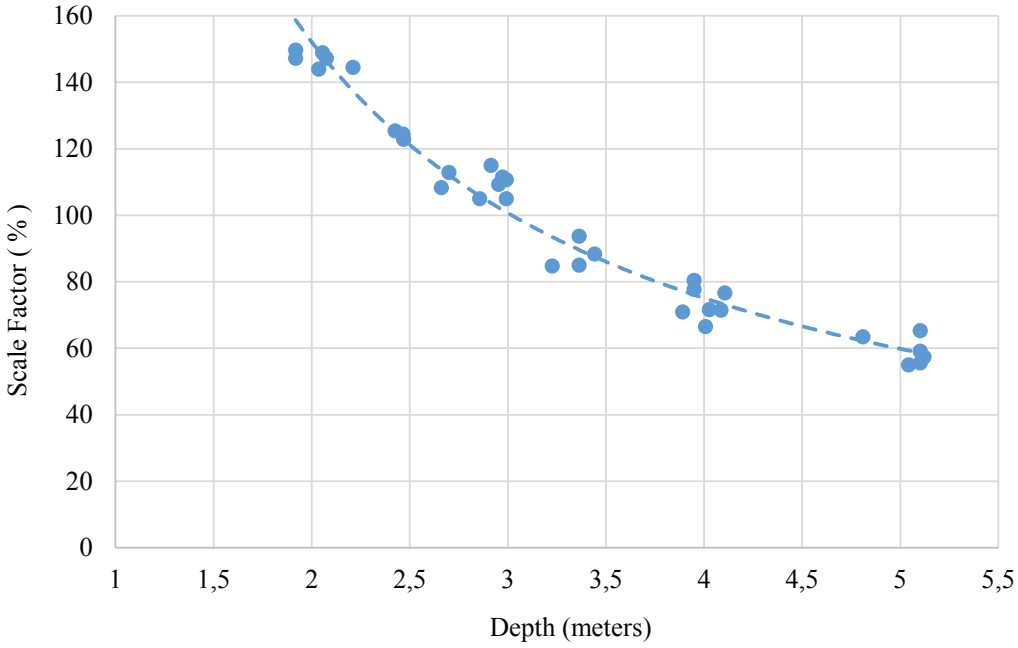


Figure 5.5: Graph representing the values for the scale factor and respective trend line.

Figure 5.5 shows the obtained results, where the xx axis represents the depth of the ROI used to fit the template, and the yy axis the scale factor, in percentage, needed to perform the correct resize for the best possible fit of the template. To calculate the scale factor for any depth, a trend-line was obtained in the form of a potential equation, given by Equation 5.2, where s is the scale factor by which the height and width of the ROI should be multiplied, d is the depth of the ROI obtained by Equation 4.10, and ρ is a fine-tuning parameter used to improve the fit for different templates using the same equation.

$$s = 308.1 \times d^{-1.018} + \rho \tag{5.2}$$

5.2.2 Template Fitting

This stage details one of the most important features on the proposed system, the template matching using part of the Perfect Match algorithm [35] proposed by Lauer, M., et al., first used on the former robotic soccer MSL team, Brainstormers Tribots, and currently used on the CAMBADA robotic soccer team, of the same league, as a self-localization algorithm, showing a high precision, robustness and computational efficiency.

The Perfect Match algorithm is used as "an efficient numerical approach to find the locally best match between the camera image and the model of the field" [35]. As far as the authors' concern, this is the first time this algorithm is used, not as a localization method, but instead as a computer-vision application, specialized in people detection. To perform the template matching a gradient descent technique is used to minimize the matching error. The RPROP algorithm [36], proposed by Martin Riedmiller in 1993, differs from other gradient descent algorithms since the calculations performed by it do not use the value of the derivative, but instead the temporal behaviour of its sign. This allows for a quicker convergence of the best position for the template, that generates the least matching error.

To perform the matching, the algorithm is supplied with a DT of the contour of the ROI, as the ones presented in Figure 5.2, as well as the templates to test, already resized to the appropriate size depending on the depth of the ROI. The aim of the RPROP algorithm is then to position the template over the DT on the position that generates the lowest error, which will be considered as the best local match.

The used images have two dimensions, therefore the template must slid in two directions, X and Y. To do so the gradient of the DT, in respect to X and Y, is calculated using a Sobel operator with a 3×3 kernel. Besides the translation, another transformation performed by the algorithm, which proved to be important for this project is rotation. By allowing the template to rotate, the classification algorithm becomes more robust, specially in cases where the person is sideways and different people tend to present different postures.

An important part of the algorithm is the initial position, due to the fact that the algorithm finds local minimums for the error, not necessarily the lowest possible error. The solution applied by Lauer, M., et al. is to calculate the error for several points, aligned on a grid in a random manner, and selecting the one with the lowest error as the starting point for the match. For the people classification problem, some characteristics of the human shape can be used to our advantage, such as the fact that if we assume that the person is in an upright position or similar, the head will always be close to the vertical center axis of the body and will be the topmost part.

To calculate the best starting position, the bounding box of the ROI is divided in vertical slices and the number of occupied pixels for each slice is counted. The template's initial X position will be the center of the most occupied slice, while the Y position will always be 0, because in general the head is the highest part of the body.

The number of slices should always be an odd number since this allows for a slice to always be present in the center of the ROI, which is the most probable place for the person's head to be. Several tests were performed, and 5 slices presents the best relation between distance travelled by the template, from the start to the end position, and the amount of calculations necessary to compute the most occupied slice. Figure 5.6 shows some examples of ROIs, with slices delimited by dashed lines, with the resulting start position is marked by a blue circle. It is important to notice that the coordinate $(0, 0)$ corresponds to the top-left corner of the bounding box.

Situations where the head is not the topmost part of the body take place when the user has one or both arms up. No solution can be presented for extreme cases when the arms touch the head, because the system requires total vision of the head and partial vision of the shoulders.

This initial position is used as an anchor point for the first iteration of the algorithm. Let (p, θ) be a pair of possible position $p = (p_x, p_y)$, and rotation θ, for the anchor point s in the global coordinate system. Let t be a list of templates $t_1, \ldots, t_j$, where j is the total number of

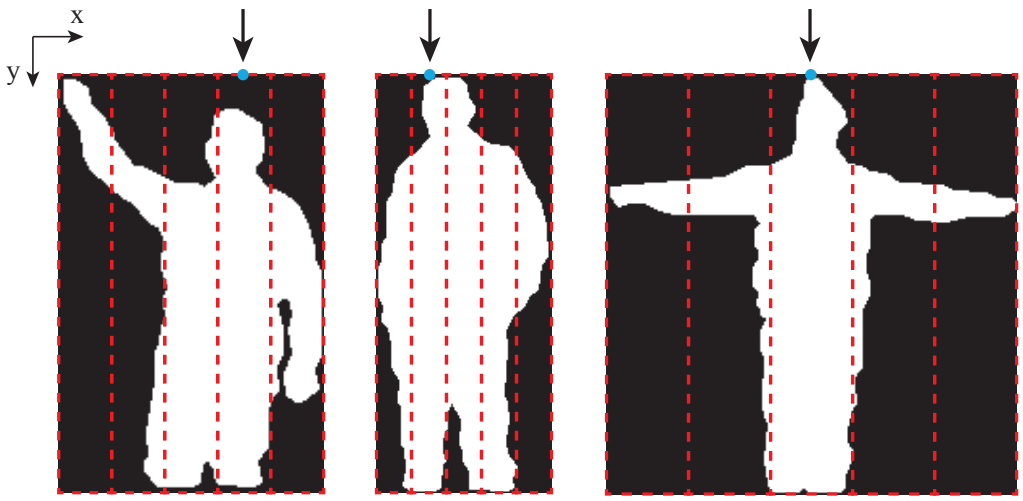


Figure 5.6: Images divided in 5 vertical slices, and starting points marked by an arrow.

templates, and from each template, a list of points is retrieved from its lines, as observations $o_1, o_2, \ldots, o_n$. The observations bare coordinates relative to the template. In order to perform the matching between the template points and DT positions, these have to be converted to global coordinates, using Equation 5.4.

$$\begin{array}{lcl} \mathcal{T} & : & R^3 \rightarrow R^2 \\ o & : & (x, y) \\ s & : & < p_x, p_y, \theta > \end{array} \tag{5.3}$$

$$\begin{array}{rcl} \mathcal{T}(s) & = & \begin{bmatrix} s_{p_x} \\ s_{p_y} \end{bmatrix} + \begin{bmatrix} \cos(s_\theta) & -\sin(s_\theta) \\ \sin(s_\theta) & \cos(s_\theta) \end{bmatrix} \times \begin{bmatrix} o_x & o_y \end{bmatrix} \\ \mathcal{T}(s) & = & \begin{bmatrix} s_{p_x} + o_x \cos(s_\theta) - o_y \sin(s_\theta) \\ s_{p_y} + o_x \sin(s_\theta) + o_y \cos(s_\theta) \end{bmatrix} \end{array} \tag{5.4}$$

Figure 5.7 shows how changing the anchor point s, top-left corner of the template's bounding box, and applying a transformation to the template points with Equation 5.4, it is possible to traverse the template using global coordinates.

By positioning these points over the DT it is possible to directly retrieve the error associated with each point, as the DT can be used as an occupation map, where each pixel's value presents the distance to the closest point of the ROI's contour.

The characteristics of Lauer, M., et al. application are quite different from ours. In [35] the Perfect Match is used as a localization algorithm provided with omnidirectional pictures, from where observations are taken from visible field lines. Due to the high distortion of the lens, and because of the vibrations the camera suffers when the robot is moving, the lines in the image appear distorted and blurred, generating outliers. To resolve this Lauer, M., et al. uses, not a standard error function $e \mapsto \frac{1}{2}e^2$, but instead a custom function more robust to outliers $e \mapsto 1 - \frac{c^2}{c^2+e^2}$. An advantage of the processing performed in order to obtain ROIs (see chapter 4) is that each retrieved ROI masks only one connected component. This means that every point in the ROI belongs to the object masked by it, no outliers exist (see Figure 5.8,

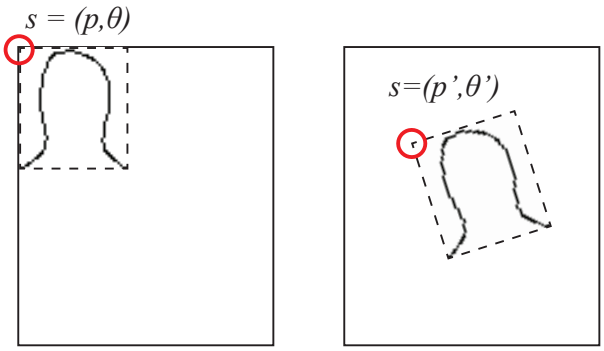


Figure 5.7: Example of different positions for the anchor point of the template.

---

c) to *f)*). This makes it possible to use the DT as an error function (see Figure 5.8, *g)* to *j)*), and fetch the error value directly from the DT for a given point.

To calculate the matching error $\mathcal{E}_t$ for a given template t, a sum of the values present in the DT matrix $\mathcal{D}$ is performed, for positions coincident with the observations, transformed by $\mathcal{T}$ for a given anchor point s. Equation 5.6 shows how the error is calculated. If, in position s, all template points perfectly match with the contours of the ROI, the resulting error is 0.

$$\begin{aligned} \mathcal{D} &: R^2 \rightarrow R \\ \mathcal{E} &: R \rightarrow R \end{aligned} \tag{5.5}$$

$$\mathcal{E}_t = \sum_{i=1}^{n} \mathcal{D}(\mathcal{T}_{o_i}(s)) \tag{5.6}$$

Due to the nature of the match it does not make sense for points outside of the image to be tested, however, simply ignoring the error present in these points would lower the matching error incorrectly. Therefore points that go outside the limits of the DT are still tested, but their position is clamped to the map's maximum value for width or height.

Different templates have different shapes, hence a different number of points. So for templates with fewer points are not benefited, the matching error obtained from Equation 5.6 is normalized, dividing the resulting error by the number of points of the template.

Because the value for each point of the DT is the distance to the closest occupied pixel, for ROIs of smaller size the resulting value in each point will never be higher than the diagonal of the ROI's bounding box, which results in low matching error even in cases where the template presents a bad fit. To be able to identify these cases another factor is taken into account when judging the final score for the classification, which is the variance of the matching error. Equation 5.7 shows how to obtain the variance for $\mathcal{E}$, where n is the number of points in the template.

$$\sigma^2_{\mathcal{E}_t} = \sum_{i=1}^{n} \frac{(\mathcal{D}(\mathcal{T}_{o_i}(s)) - \overline{\mathcal{E}_t})^2}{n} \tag{5.7}$$

By analysing $\sigma^2_{\mathcal{E}_t}$ it is possible to understand if the error of each point considerably varies. This measure is as important as the matching error $\mathcal{E}_t$.

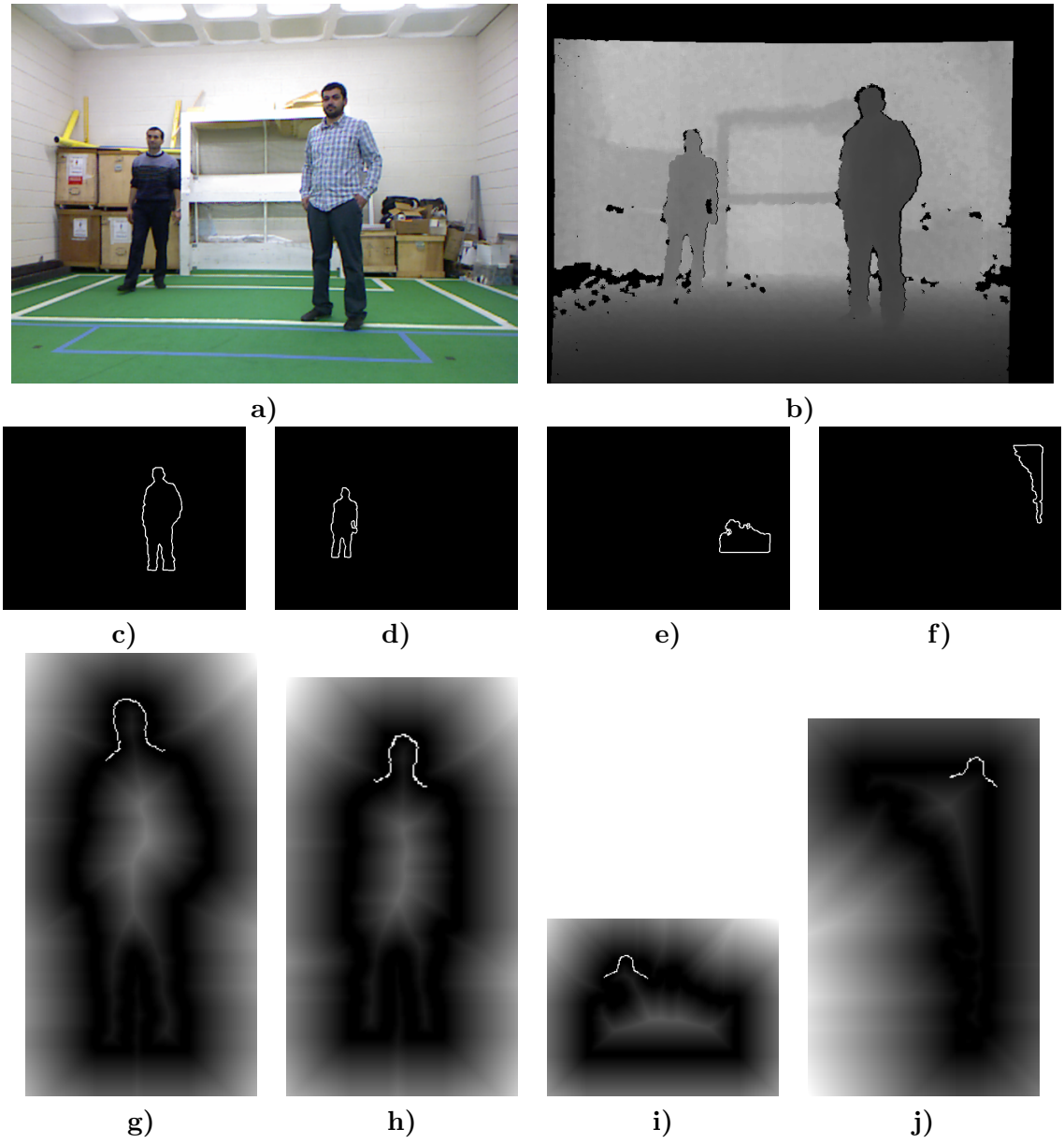


Figure 5.8: Template fitting process. Kinect RGB capture in *a)*, Depth capture in *b)*, ROI's contours from *c)* to *f)* and DT of each ROI, with template overlapped in the best position obtained by the algorithm, from *g)* to *j)*.

Figure 5.8 illustrates a typical template fitting process. The color and depth captures of the Kinect cameras are shown in *a)* and *b)* respectively. Images *c)* to *f)* present the contours of ROIs 1 to 4 respectively, obtained from the depth capture shown in *b)*, in which the first two are humans and the last two are not humans. Images *g)* to *j)* show the resulting DT obtained from each ROI's contour, normalized, to make their visualization possible, with templates drawn over. Every template was tested in each DT, the one shown in the images being the one that presents the lowest matching error and therefore the one chosen as the most correct

for that situation. For images *g)* and *h)* the front template (Figure 5.4 *a)*) presents the best fit, and for images *i)* and *j)* the back template ((Figure 5.4 *d)*) presents the best fit.

It is possible to observe some important aspects in this figure: first, the size and rotation of each template are optimized for the corresponding DT; second, notice how ROIs that do not present a human shape tend to facilitate the fitting of the backwards template due to its simpler shape.

An important parameter, which affects greatly the fitting process, is the thickness of the line that makes up the contour of the ROI, used to generate the DT. By assigning a low value, such as 1 pixel wide, the algorithm presents a greater difficulty when trying to fit the template. If the thickness is increased, the fitting becomes easier in every ROI, however, because the shape of the ROI is not altered, human ROIs tend to benefit more from this facility than non-human ones. Tests with different values for this parameter are presented in chapter 7.

Using ROIs 1 to 4, presented in Figure 5.8 *c)* to *d)*, as an example, Table 5.1 presents the values for $\mathcal{E}$ and $\sigma_{\mathcal{E}}$ of each.

Region of Interest	$\mathcal{E}_t$	$\sigma^2_{\mathcal{E}_t}$
ROI 1	0.046776	0.042483
ROI 2	0.027286	0.025313
ROI 3	0.887823	1.297435
ROI 4	2.238443	3.492047

Table 5.1: Error ($\mathcal{E}_t$) and error variance ($\sigma_{\mathcal{E}_t}$) values for ROIs presented in Figure 5.8.

It is possible to see in Table 5.1 that ROIs 1 and 2, human ROIs, present very low errors and variance, while ROI 4, a non-human ROI, presents the highest error and variance. The importance of the variance can be noticed in ROI 3, where its matching error is not far from the errors of ROIs 1 and 2, know to be human. However, if the variance is observed it can be seen that it is higher than ROIs 1 and 2. This reflects the fact that the template crosses zones in the DT with high error, which indicates a bad fit, even if it is the best one calculated.

The information provided by the matching error and variance is only used for classification purposes. To propel the template through the DT until this reaches its best position, a gradient descent technique is used, the RPROP algorithm [36]. The gradient of the error is the sum of the gradient for the observations o, in a given state s. Let D be the DT array with no closed formula, and T the transformation for the observations as shown in Equation 5.4. Equation 5.8 presents the equation necessary to calculate the gradient of the error, $\nabla_s \mathcal{E}_t$, in the three spaces, X, Y and θ.

$$\begin{aligned}
\nabla_s \mathcal{E}_t &= \textstyle\sum_{i=1}^{n} \nabla_s \mathcal{D}(\mathcal{T}_{o_i}(s)) \\
&= \textstyle\sum_{i=1}^{n} \nabla_{\mathcal{T}_{o_i}(s)} \mathcal{D} \times \nabla_s \mathcal{T}_{o_i}(s) \\
&= \begin{bmatrix} g_x & g_y \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -o_x \sin(o_\theta) - o_y \cos(o_\theta) \\ 0 & 1 & o_x \cos(o_\theta) - o_y \sin(o_\theta) \end{bmatrix} \\
&= \begin{bmatrix} g_x & g_y & g_x \left(-o_x \sin(s_\theta) - o_y \cos(s_\theta)\right) + g_y \left(o_x \cos(s_\theta) - o_y \sin(s_\theta)\right) \end{bmatrix}
\end{aligned} \tag{5.8}$$

In order to move the template the algorithm needs to know which direction is the correct, positive or negative for X, Y and θ. Some gradient descent algorithms use the value of the

gradient to determine if the the function is close to a local minimum or maximum, however the RPROP algorithm uses not the value, but the signal of the gradient.

To determine the movement direction, the gradient value for the current state $\nabla_s \mathcal{E}_t$ is multiplied by the gradient value in the previous state $\nabla_{s'} \mathcal{E}_t$. If the result is negative it means that the gradient changed behaviour and, therefore, the anchor point used to move the template is close to a local minimum for the error. This procedure is repeated for gradients in all dimensions, $\nabla_x \mathcal{E}_t$, $\nabla_y \mathcal{E}_t$ and $\nabla_\theta \mathcal{E}_t$, which allows for horizontal, vertical and rotative movement of the template. Moreover, the step size γ, by which the template is moved, increases 20% each time the gradient maintains its signal and it decreases 50% when the signal changes, as show by Equation 5.9.

$$\gamma_i = \begin{cases} \gamma_{i-1} \times 1,2, & \nabla_s \mathcal{E}_t \times \nabla_{s'} \mathcal{E}_t > 0 \\ \gamma_{i-1} \times 0,5, & \text{otherwise} \end{cases} \tag{5.9}$$

For each iteration of the RPROP algorithm, the anchor of the template, s, is moved by γ. The matching error is calculated for different anchor points during these iterations, being $\mathcal{E}_t$ equal to the smallest error achieved.

The final matching error assigned to a ROI comes from the template that generates the lowest error when compared against the DT in the best achieved position. Letting $\mathcal{E}$ be the best matching error, it can be obtained by Equation 5.10.

$$\mathcal{E} = \min_j(\mathcal{E}_{t_j}) \tag{5.10}$$

5.3 Region Classification

Having more than one measure to characterize an object, in this case the matching error $\mathcal{E}$ and its variance $\sigma^2_{\mathcal{E}}$, makes it necessary to relate their two values under a single result. This result can be seen as a confidence level in an object being human, the higher the value, the better the fit of the template over the contour of the ROI, in the best position achieved by the RPROP algorithm, and therefore the higher the probability of the object being a person.

Due to the nature of the values, the best formula that one can use to relate these two value is a normalized Gaussian Distribution. Normal, or Gaussian, distributions are often used for analysis of random variables whose distributions are not known. Let $\mathcal{S}$ be a Gaussian center around zero, $\mathcal{E}$ the matching error and $\sigma^2_{\mathcal{E}}$ the according variance. The confidence level can be calculated using Equation 5.11.

$$\mathcal{S}(\mathcal{E}, \sigma^2_{\mathcal{E}}) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{E}}}} \times e^{-\frac{\mathcal{E}^2}{2\sigma^2_{\mathcal{E}}}}, \text{ where } \mathcal{E} \in R^+ \text{ and } \sigma^2_{\mathcal{E}} > 0 \tag{5.11}$$

Using the values present in Table 5.1, the score of each ROI is presented in Table 5.2.

By judging the confidence level of each ROI, it is possible to classify it as human or not-human. Due to the flood fill technique used to obtain the ROIs, regions whose contours are not easily distinguishable present a great variation in shape due to overflows. An example of this can be seen in Figure 5.8 *f)*, where the wall is only partially segmented and its contour does not respect any specific shape, presenting great variations in consecutive frames. Although in most frames the shape of this ROI does not generate a high score, in some, its score is high enough to be considered human, causing intermittent human detections.

Region of Interest	$\mathcal{E}_t$	$\sigma^2_{\mathcal{E}_t}$	$\mathcal{S}$
ROI 1	0.046776	0.042483	1.886339
ROI 2	0.027286	0.025313	2.470858
ROI 3	0.887823	1.297435	0.258490
ROI 4	2.238443	3.492047	0.104182

Table 5.2: Table presenting the confidence for the classification of ROIs presented in Figure 5.8.

So as to provide greater robustness to the system, avoiding false positives (ROIs erroneously classified as human), the algorithm does not perform classification based on only one sighting, but instead on a sighting mechanism throughout several frames.

Algorithm 5.1 presents this classification mechanism based on multiple sightings of the same ROI. In it $\mathcal{S}_r$ represents the confidence of a ROI r being human, l is a location in global coordinates (x, y, z), and c_l the number of consecutive sightings on a given location.

Several parameters are used to control the classification: υ is the minimum confidence for a ROI to be considered human, α is the value by which a consecutive sighting is incremented, while α' is the value by which is decremented, ν is the maximum Euclidean distance between two locations for them to be considered as part of the same sighting, and κ is the minimum number of consecutive sightings before a ROI is considered human.

Algorithm 5.1 Algorithm used to classify a ROI as human or not-human.

ClassifyROI
  l' = closest sighting to the ROI in l_r
  if $\mathcal{S}_r > \upsilon$ **then**
    if $l_r - l' < \nu$ **then**
      Update $l' = l_r$
      $c_{l'} = c_{l'} + \alpha$
    else
      Create a new sighting in l_r
      $c_{l_r} = 1$
      $l' = l_r$
    end if
    if $c_{l'} > \kappa$ **then**
      Classify r as human
      End
    end if
  else
    if $l_r - l' < \nu$ AND $c_{l'} > \kappa$ **then**
      Update $l' = l_r$
      $c_{l'} = c_{l'} - \alpha'$
      Classify r as human
      End
    end if
  end if
  Classify r as not-human

The main cause for incorrect classification are ROIs whose contours considerably vary, causing some of them to approximate to a human shape for brief moments. The previous algorithm takes the classification through confidence even further, by penalizing ROIs that do not present a continuously high confidence level and rewarding those which have a high confidence throughout consecutive frames.

Adjusting υ, ν, and κ, it is possible to decrease the number of false positives, however, by doing so, the probability for a ROI to be classified as human also increases. κ affects the detection speed of the system by controlling how many high confidence detections are need for a ROI to be considered human. α and α' controls how the number of consecutive sightings grows and decreases and, therefore, how flexible the classification is, for example, when the person is in a certain position where the head can not be completely seen. Because the values for these parameters affect severely the human detection capabilities of the system, different values were tested for each parameter, being the results presented in chapter 7.

Chapter 6

Human Tracking

The second capability of the system proposed in this work is the ability to track detected persons. Tracking refers both to the assignment of a different ID to each new person detected and to the monitoring of each person's pose in every frame. People detection is essential for individual human-robot interaction and the ability to identify a certain individual among others allows for more complex and specialized applications.

Tracking a single human object in an image can be a task as simple as recording the ROI's centroid coordinates throughout several frames, since, if it is assumed that there is only one person in the frame at all times, one can expect that the ROI belongs to the same person throughout the entire capture. On the other hand, tracking multiple persons is more complicated, because it means that the system must know who is who, throughout the presence of each person in the field of view of the camera. Using the location of each person to determine who is who in an image is not a valid method, due to occlusions, people crossing paths and the entering and exiting of different people in the scene.

As mentioned before, the human body presents several degrees of freedom and because the depth image only provides information on an object's shape, comparing different peoples shapes does not guarantee good results, both because the same person can present very different shapes (ex. if his arms are up or down) and because different people can present similar shapes when in a relaxed pose.

This triggers color images as a complementary source of information. In real cases, searching for a person among a cluttered environment, whether the space is occupied by non-human objects or he/she is in the middle of a crowd, comes down to the searching for the person's cloth or hair colors as the most distinguishing visible feature. The proposed system mimics this behaviour and is able to distinguish persons by the colors present in their figure, being able to consistently distinguish different persons present in the same capture, if the lighting conditions do not change drastically, even after they exit and re-enter the camera's field-of-view.

The diagram depicted in Figure 6.1 performs a quick overview over this process. The ROIs classified as Human, by the People Classifier in the previous stage, are given a unique ID and their face is recorded if its the first time the ID has been assigned. Furthermore, both the pose and location of each person is estimated.

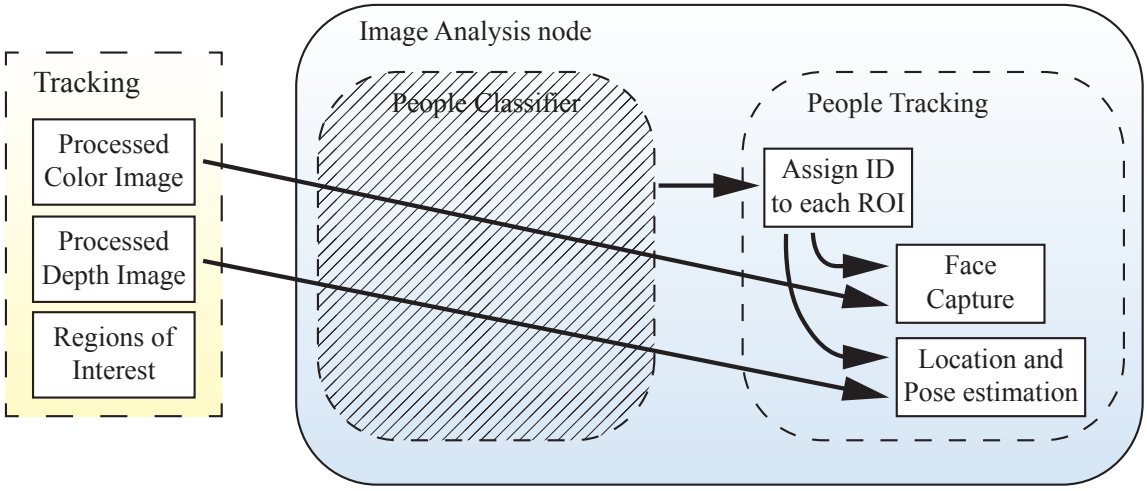


Figure 6.1: Diagram detailing the tracking process for ROIs classified as human.

6.1 Histogram Comparison

Histogram comparison is a widely used method in several areas and Mean shift, applied to Computer Vision, resorts to this comparison technique to implement an iterative object search algorithm in an image. This algorithm is used by authors such as Ikemura, S., et. al. in [6] and requires the object to be visible before the tracking begins, so that its back projection is calculated. The back projection is then used to perform an iterative search, using the histogram of both, the object and of a certain region of the image.

While studying the Mean Shift algorithm an important conclusion was ascertained. At this stage, two important informations about objects classified as human are know. Both the position and the limits of the object are determined by its ROI, generated by the algorithms presented in chapter 4. Therefore the objective of the identification algorithm should not be to search for a person in the entire image, but rather to relate the same human ROI throughout different frames.

Using histogram comparison to perform this tracking, instead of the coordinates of the person, adds robustness to the proposed system, because it also resolves problems such as identification after occlusions and eventual detection flaws.

The system starts by calculating the histograms for new human ROIs. At this point two important attributes should be taken into account: the color space used to calculate the histogram and the region of the color image from which data is going to be retrieved.

Several color spaces were considered to perform this comparison, such as RGB, CMYK and HSV. The last was chosen due to the nature of the channels it is composed of (see Figure 6.2). The H channel stands for *Hue* and is capable of differentiating color through a single channel, instead of a combination of channels such as in spaces like RGB and CMYK. The S stands for *Saturation* and quantifies, as the name says, the color saturation, where lower values move the color into a gray level and higher values saturate the color. Finally, channel, V, stands for Value, and deals with the brightness of the color, where lower values darken the color and higher values brighten it.

The second important aspect to consider is the area of the image from which the histogram

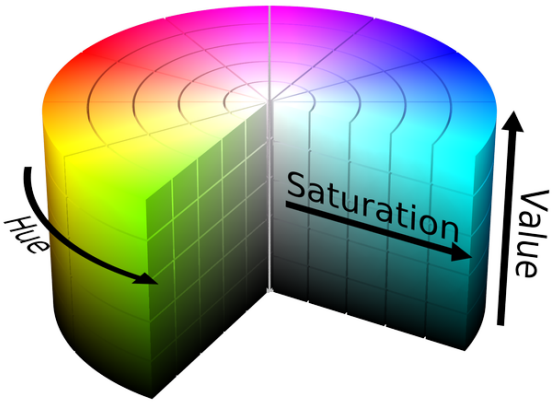


Figure 6.2: Representation of the HSV color space.

is going to be calculated. Initially the histogram was calculated using the region of the color image masked by the entire ROI, however, after some tests, the comparison of the histograms for the same person in different frames produced inconsistent results. This is caused by colors present near the limits of the contour of the person but that does not belong to her, since the register of the depth image (from which the contour is obtained) over the color image might not be perfect. Therefore, the solution presented for this problem is to shrink the area used to generate the histogram. Furthermore, the histogram of the ROI is normalized, so that the size of the region does not affect the comparison on subsequent appearances of the same person at different depths. Figure 6.3 shows the region of the color image masked by the original ROI, in *b)*, and *c)* the shrunk region used in the comparison.

a) **b)** **c)**

Figure 6.3: Shrunk region of the color image, used for histogram comparison. In *a)* the color image capture, in *b)* the original ROI and in *c)* the shrunk region.

The comparison is performed individually for each channel, between existing tracked ROIs and ROIs present in the current frame. The correlation in done separately for each channel so that different weights can be used when calculating the final value for the likeness between two tracked human ROIs. For example, the Hue channel will have the highest weight of the three channels, because it can represent color on its own and since it is the most robust to changes in illumination of the three.

The metric used to perform the comparison is the Chi-Square. Let $\mathcal{H}_{1_c}$ be the histogram

for a channel c of ROI 1, and $\mathcal{H}_{2_c}$ the same but for ROI 2 and b the total number of bins. The difference between these two ROIs can be calculated according to Equation 6.1, as d. For histograms exactly equal the result will be 0, while higher results mean a worse matches.

$$d(\mathcal{H}_{1_c}, \mathcal{H}_{2_c}) = \sum_{i=1}^{b} \frac{(\mathcal{H}_{1_c}(i) - \mathcal{H}_{2_c}(i))^2}{\mathcal{H}_{1_c}(i)} \tag{6.1}$$

The final difference between histograms for two ROIs can then be obtained by adding the difference from each channel. Let d_h, d_s and d_v be the difference between channels Hue, Saturation and Value respectively, and ω_h, ω_s and ω_v the weights for corresponding channels. Equation 6.2 shows how the final value between ROIs is computed, as $\mathcal{D}$.

$$\mathcal{D} = d_h \times \omega_h + d_s \times \omega_s + d_v \times \omega_v \tag{6.2}$$

6.2 Individual ID Assignment

Assigning an identification number to each human ROI can be seen as an assignment problem where no repetition is allowed. Among the most used solutions for this type of problems, the Hungarian method is one of the most popular optimization algorithms available. One of the requirements for this algorithm is a non-negative square matrix ($x \times n$), where the element in the *i-th* row and *j-th* column represents the cost of assigning an *action j-th* to the *i-th object*.

The study of the Hungarian method led to the arrangement of the problem in question as a matrix, which simplifies its understanding. Let the rows of the matrix hold the histograms of human ROIs detected in the current frame and the columns hold the histograms of human ROIs tracked from previous frames. The cell in the *i-th* row and *j-th* column, holds the difference between histograms, computed as detailed in the previous section.

However, a problem arises when applying the Hungarian method to identification assignment: the dimensions of the matrix. For example, in the first captured frame, one person is detected, and because there are no active tracks yet, the result is 1×0 matrix. Furthermore, cases were the number of persons detected in one frame is different from the next frame, will always generate a non-square matrix. This makes the fulfilment of one of the requirements of the Hungarian method impossible. Moreover, if the difference between histograms is higher than a certain value, it should be considered as a newly tracked ROI. Due to these incompatibilities a new optimal assignment algorithm is proposed.

Algorithm 6.1 represents how the identification procedure works. In it, C is a comparison matrix, organized as explained before, and A is the ROI assignment vector, used to hold the track ID. A number of parameters is used to control the operation: N is total number of humans present in the current frame, E the maximum allowed comparison error for two ROIs to be considered the same and P the number persons tracked so far.

Because in the Chi-Square square metric a lower value is assigned to similar histograms, the first step is to search for the minimums in each column and use its index as the first choice for each human ROI. If the comparison error in this element, although being the lowest, is still too high to be correctly considered a good match, then the value -1 is assigned to this ROI. When a ROI is assigned -1, the algorithm creates a new tracking ID in the end of the algorithm. In cases where P equals 0, meaning no persons are being tracked, the persons detected in the frame are all assigned new IDs.

Algorithm 6.1 Algorithm used to assign a track ID to each ROI.

OptimalAssign

Require: $N > 0 \wedge P > 0$
  $A_i \leftarrow -1, \forall i \in [1, N]$
  $T_i \leftarrow FALSE, \forall i \in [1, N]$
  while assignment not complete **do**
    for all $r \in [1, N]$, where $T_r =$ FALSE **do**
      if $\min_c(C_{r,c}) < E$ **then**
        $A_r \leftarrow c$
      else
        $A_r \leftarrow -1$
      end if
      $T_r \leftarrow TRUE$
    end for
    for all $i \in [1, N]$ **do**
      for all $j \in]i, N]$ **do**
        if $A_i == A_j \wedge A_i > -1$ **then**
          $k = \arg\max_{i,j}(C_{i,A_i}, C_{j,A_j})$
          $A_k \leftarrow -1$
          $T_k \leftarrow$ FALSE
          $C_{k,A_k} \leftarrow C_{k,A_k} + E$
        end if
      end for
    end for
    if $T_i =$ TRUE, $\forall i \in [1, N]$ **then**
      assignment complete
    end if
  end while

In the best possible case, one iteration over the comparison matrix is enough to generate associations without repetition, meaning that every ROI has its own individual track ID and so the algorithm converges. In cases where more than one ROI chooses the same track ID (except for ID -1), the algorithm proceeds as follows: from the ROIs that have chosen the same track ID, only the one with the lowest comparison error will maintain its selection, while others will discard it, and the comparison error present in C_{k,A_k},is incremented by E, so as to force the ROI to choose a different track ID in the next iteration of the algorithm.

When a ROI is associated with a track ID, the histogram from the track is update. By updating the histogram at each frame, when a new ROI is associated, the algorithm allows for subtle changes in the histogram, due to different light conditions or shadows. This mixture is performed by applying a weighted sum for each channel, from the ROI's current histogram and the histogram of the track, from the previous frame.

An example of histogram comparison and respective track ID association can be seen in Figure 6.4. The diagram shows images captured at three different instants of time, t_x, t_y and t_z, on the left side, and the resulting *comparison matrix* followed by the *assignment vector* for each frame, on the right side. At t_x a frame is captured, where one person is detected and, because there are no tracks yet, the ROI for this person is associated with the ID -1, which forces the system to create a new tracking ID. Immediately following this capture, at frame t_{x+1} there is one active track and, because the comparison result is very low, the ROI is associated with the track ID recently created. At frame t_y the situation is similar, but now there are two human ROIs in the image and only one active track, generating a 2×1 *comparison matrix*. The ROI with the lowest comparison error is assigned to the existing track ID while the other will receive a new track ID in the next frame, t_{y+1}. The rightmost person, although being completely capture in the color image, has part of the head outside of the depth image due to the image registration, and therefore it is still not detectable in this frame. Finally, in the last captured frame, t_z, three persons are detected. Notice how, in frame t_{z+1}, the difference between comparison errors $C_{1,2} = 0.88$ and $C_{1,3} = 12.57$, and $C_{3,2} = 10.13$ and $C_{3,3} = 3.11$, is considerably lower than between $C_{2,1} = 0.93$, $C_{2,2} = 75.21$ and $C_{2,3} = 46.32$. This is due to the colors present in each person's clothes: the leftmost and the right most persons are both wearing brown shirts, the only difference being that one is lighter than the other, while the person in the middle stands out from the rest because he is wearing darker clothes.

6.3 Location and Pose Estimation

The location and pose estimation for each tracked person represents valuable data for future applications intended for a service robot such as the CAMBADA@Home.

The location of a person in global coordinates can be obtained with the aid of geometric functions provided by the ROS middleware and the global localization of the CAMBADA@Home robot, provided by a self-localization system previously developed for this platform. The global coordinates of the robot are constantly being published by the localization algorithm running in the background. Knowing the robot's global position, and the position of the Kinect relative to it, it is only necessary to apply so geometric functions, such as the one used in subsection 4.3.2 to determine the row in the image that corresponded to a certain height Z, but now applied also to X and Y coordinates, and determine the coordinates of the person relative to the Kinect. After obtaining each of these relative coordinates, all it

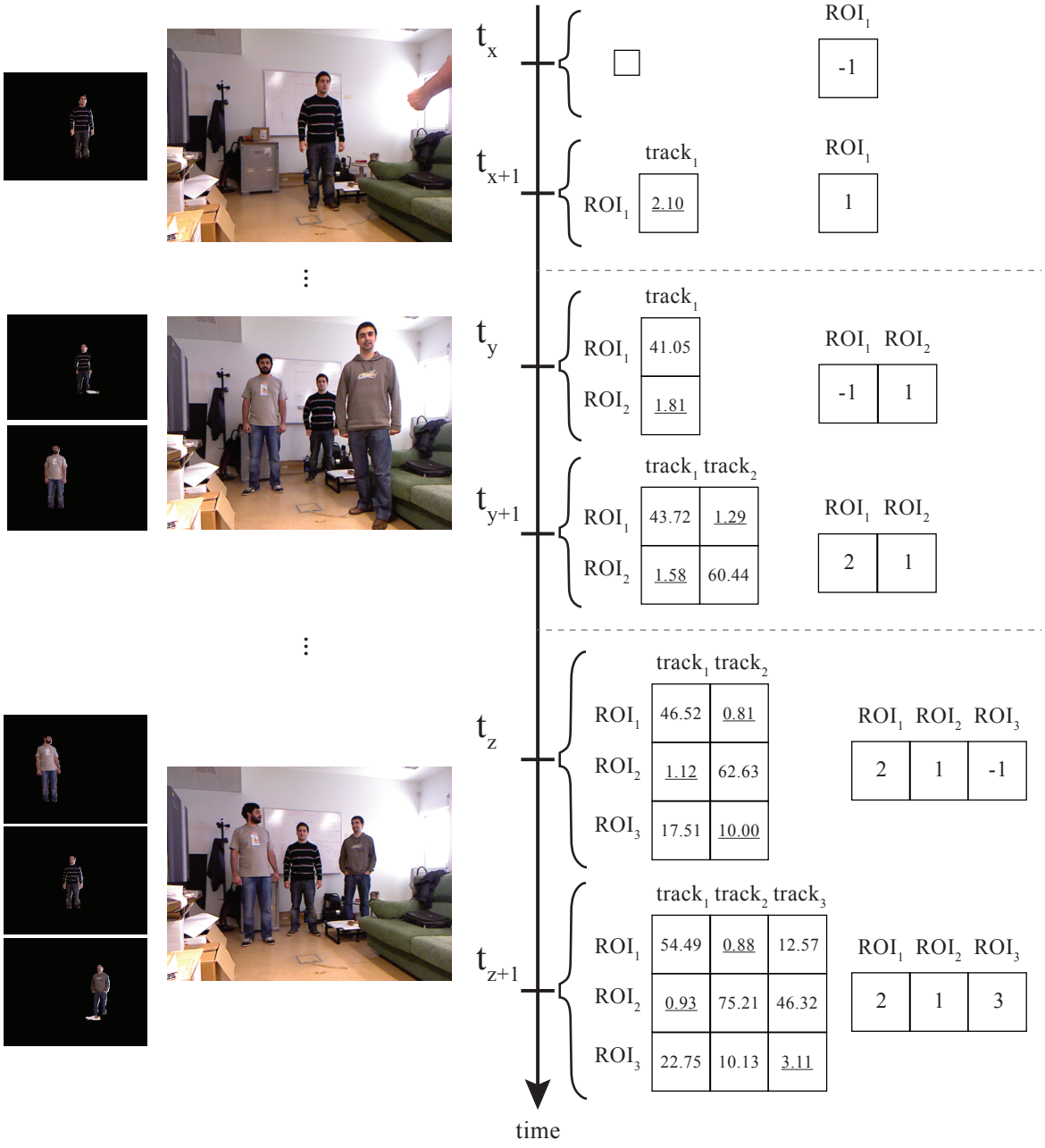


Figure 6.4: Example of the assigning process.

is needed is to sum them, obtaining the global coordinates for a person.

As for the pose of the person, it is also easy to estimate. Using different templates is advantageous for the detection process, while at the same time it allows for the pose of the person to be estimated, by relating it with the template that generated the least amount of error. To each of the templates proposed in section 5.1, a stance is associated. For the templates shown in Figure 5.4, $template_1$ in *a)*, the person is assumed to be facing towards the camera, $template_2$ in *b)* the person is facing right, $template_3$ in *c)* facing left, and for

template$_4$ in *d)* the person has his back facing the camera.

Table 6.1 shows the resulting score associated with each template, for the examples shown in Figure 6.5. The rows of this table represent the captures in the figure, while the columns represent each of the proposed templates. The underlined score is the highest in each row, identifying the template that presents the best fit for that situation, therefore indicating a pose. Also, the results for the matching of *template*$_4$ could not be obtained for the sideways stances because, after resizing the template according to the depth of the person, it presented a width higher than the one measured by the ROI.

Pose	*template*$_1$	*template*$_2$	*template*$_3$	*template*$_4$
Image *a)* (Front)	0.619197	0.462504	0.159215	0.074155
Image *b)* (Back)	0.137820	0.052670	0.052981	0.682296
Image *c)* (Left)	0.163329	0.204104	0.140372	*n.a.*
Image *d)* (Right)	0.176253	0.182221	0.282763	*n.a.*

Table 6.1: Errors for each template used for pose estimation of the captures presented in presented in Figure 6.5.

In the images presented in Figure 6.5, from *a)* to *d)*, different captures are shown covering different poses and next to each there is a text box with relevant information. Following the user ID, between curved brackets, is shown the stance. Below the user ID is the confidence of that ROI being human. Finally, in the bottom row, the position in global coordinates is shown. Furthermore, the images are presented with an overlay that darkens regions where a human is not present, leaving the remainder in normal color.

6.4 Face Detection

One of the objectives of this thesis was to perform perform people identification, so that multiple people present in the same capture can be distinguished. One method to perform this is to enable facial recognition of the persons detected. However, due to time constraints this could not be accomplished in its totality. Among other uses, the thermal image, made available by the GOBI camera, should have been used to perform facial recognition or as an alternative classification method, however, for this to happen, the thermal image has to be registered over the color image, as happens with depth image. Because the registration procedure could not be completed in time, the color image has been used instead, but only to capture the faces of detected persons.

The proposed system does not employ skin detection or any other type of facial features recognition, it uses templates that require full vision over the person's head. Taking advantage of the robustness of the matching algorithm, and the fact that, if the template is correctly positioned, it will always match the shape of the head, it is possible to segment just the face of the person, and use it to perform facial recognition.

During the participation of the CAMBADA@Home in the Free Bots Challenge, a competition at the Robótica 2013 (2013's edition of the Portuguese Robotics Open), this method was tested, proving to be effective, and capturing most of the faces of the jury in a completely automatic manner. Figure 6.6 shows some of the captured faces for people facing the camera. The capture of the face region was only performed if the template with the highest confidence was the frontal one, and the confidence level itself was above a high threshold to guarantee a

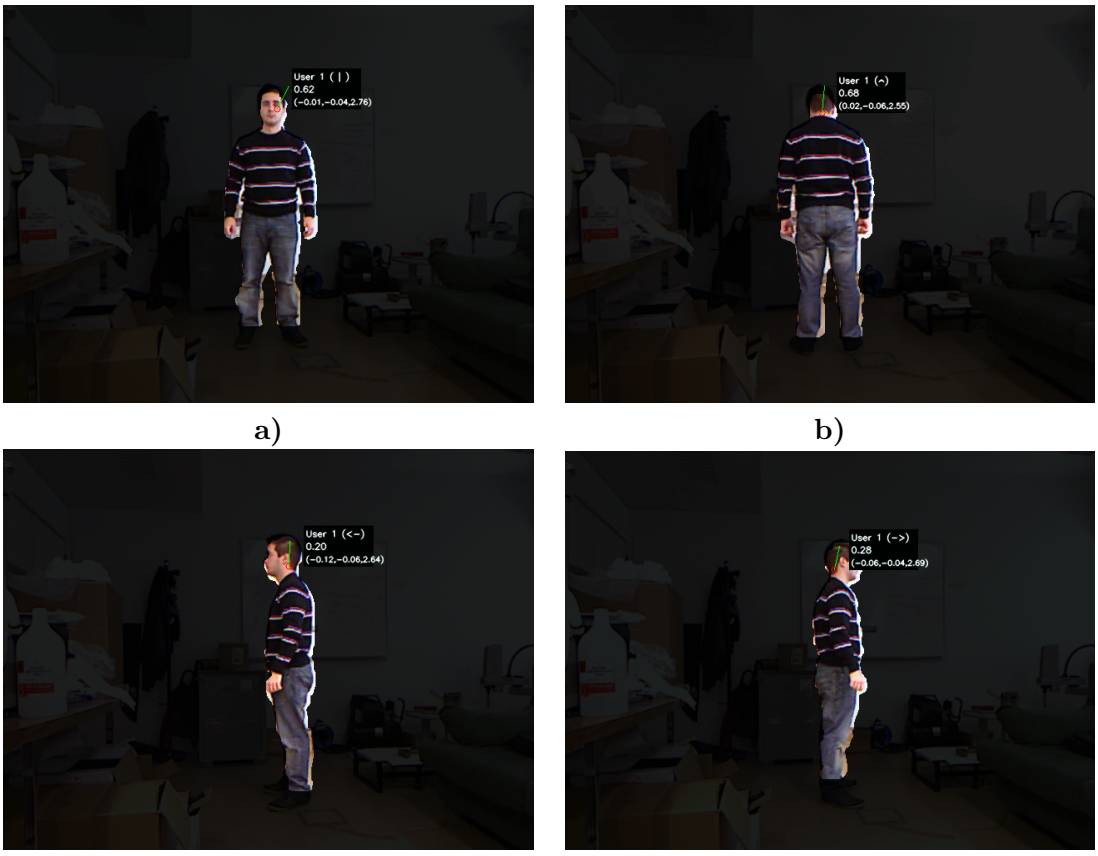


Figure 6.5: Example of poses and pose estimation.

---

correct capture. The different sizes of the images occur due to each person's distance to the camera.

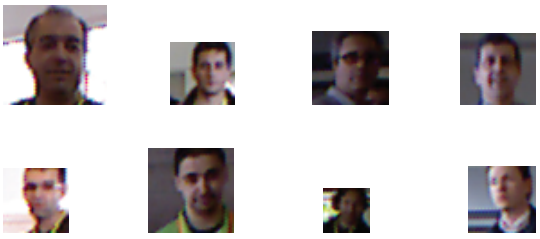


Figure 6.6: Faces automatically acquired during the Free Bots Challenge at Robótica 2013.

Chapter 7

Experimental Results

This chapter describes the experiments performed in order to test the system and obtain numeric results as to determine the system's capability of detecting and tracking people using a Kinect camera.

Some researchers provide datasets valuable as comparison tools for different systems' performances. Spinello L., et. al., provide a good dataset manually annotated, however, as with datasets from other authors, this does provide correct information necessary for tracking, as the depth image is not registered on the color image, preventing the correct functioning of the proposed system. Furthermore, one of the main characteristics of the proposed system is the detection of people while the camera is moving, therefore, experiments should be performed taking this into account.

The participation of the CAMBADA@Home in the Free Bots Challenge, a competition at the Robótica 2013 (2013's edition of the Portuguese Robotics Open), which took place in a public school in Lisbon, Portugal, provided the opportunity to perform tests in a truly unconstrained environment and use this data, as a dataset on which to perform the experiments shown in this chapter.

The results are presented in the form of tables and charts, obtained through the calculation of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Because the objective of the system is to classify objects, a classification is considered a TP when a ROI representing a person is actually classified as a person, while a TN occurs when a ROI is classified as not-human while not being a person.

7.1 System Requirements

When developing the system, one of the intentions was to create a tool for people detection as general as possible, with few restrictions.

In order to perform proper tracking, the depth image has to be registered on the color image, so that a ROI in the depth image presents an equivalent region in the color image. Also, in order to perform people detection, the head and at least partially the shoulders of the person must be visible, regardless of the person's stance, like, for example, turned sideways or even sitting down. Figure 7.1 shows an example of two captures, with registration enabled. Image *b)* shows an example of valid detection, were the head and shoulders of the person are visible, and image *d)* shows an example were, although the head and shoulders of the person are fully visible in the color image, in the depth image these features are not visible, therefore,

this is not considered to be a valid detection.

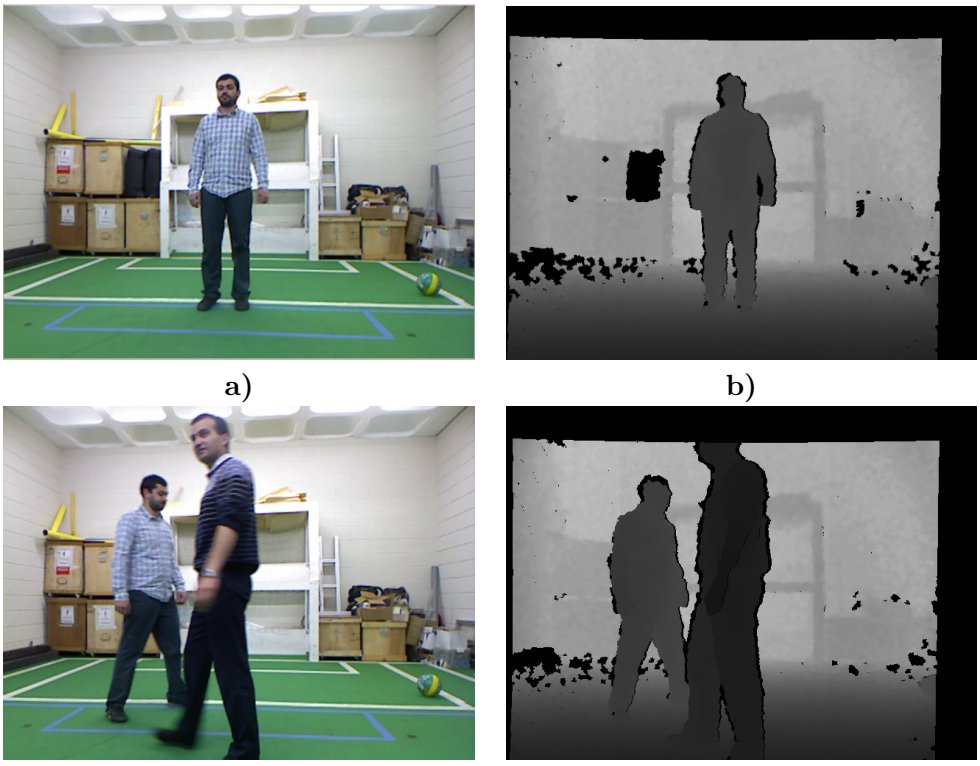


Figure 7.1: Example of valid detection, in *a)* and *b)*, and invalid detection, in *c)* and *d)*

The speed at which the system can process new image frames affects its performance, due to the need of several consecutive frames in order to classify a object as human. As the machine were the system was implemented is not capable of processing frames at 30 *Hz*, speed at which they are generated by the Kinect camera, the datasets have been replayed at a slower rate than at which they were recorded. The presented test results were obtained by replaying the datasets at a rate of 6 *Hz*.

7.2 Testing Environment

The capabilities of the proposed system were tested under two different environments, each with specific characteristics. The first dataset, named Field, was captured with the camera in a stationary position, on a spacious room, with some clutter present in the background. This environment allows for two people to be comfortably present in the same capture, while entering and exiting the field-of-view of the camera and generating several occlusions by crossing paths. The focus of this dataset is to test the performance of the system in a simple, somewhat controlled environment, while presenting two people in several stances.

This dataset features 1442 frames, with 3583 ROIs being detected by the process presented in chapter 4, and with 1639 being marked as human. Some examples of captures in the proposed dataset are presented in Figure 7.2, in chronological order.

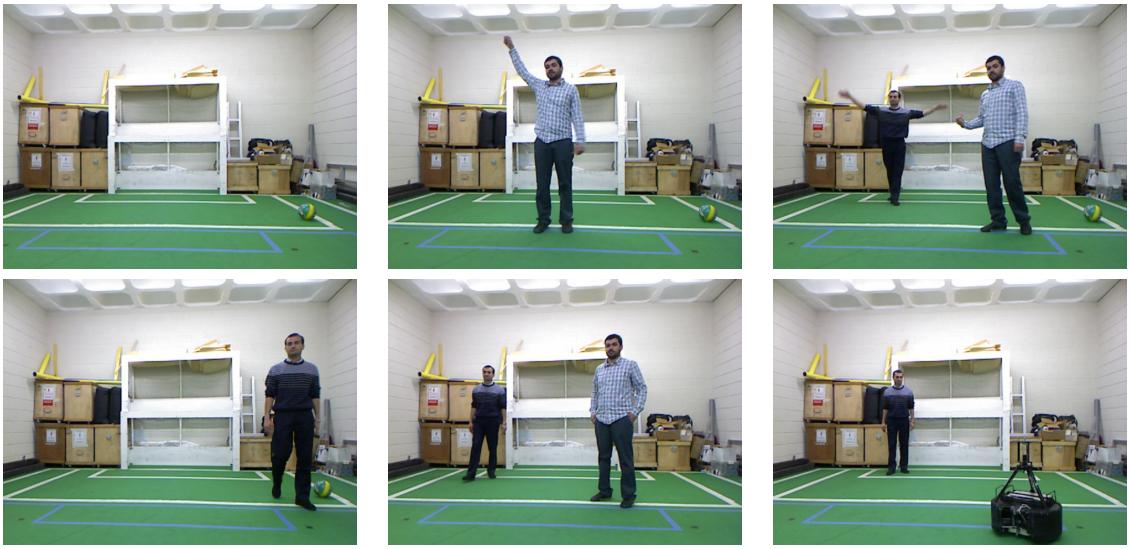


Figure 7.2: Capture examples for Field dataset

The focus of the second dataset, named Corridor, is the detection of moving people in an unconstrained environment, with the camera mounted on a robot which is performing autonomous movement. The environment of the test is a big corridor located in a High School, which was the same environment faced during the *Robótica 2013 Free Bots Challenge*, in which the CAMBADA@Home project was awarded first place, by demonstrating a service robot capable of performing autonomous movement in an open space, while detecting and tracking people it encounters along the way. This dataset is more complex than the first, as the robot itself is moving, while trying to detect passing people, and also because the natural illumination of the environment greatly affects the irregularity of the contours of most objects. As in the first dataset, there are 3 persons often entering and exiting the camera's field-of-view at different times, enabling the testing of the tracking capabilities of the system.

This dataset features 1823 frames manually annotated with the persons in scene. The number of ROIs detected throughout the capture is 5516, being 1500 of them marked as human. Examples of captures in the second dataset are presented in Figure 7.3, in chronological order.

In both datasets, for each frame, both the number of detected objects and a pixel close to the center of each person are annotated. A test is perform to determine if one of the annotated pixels is positioned inside regions delimited by ROIs classified as human, and if it is outside of a regions classified as not-human.

The depth image has a resolution of 800×600 pixels, 8-bit encoding and a gray-scale color palette, this image being the result of the process explained in section 4.1. The color image also has a 800×600 resolution, 8-bit encoding, and BGR color palette.

The results are presented in the form of a Confusion Matrix, a widely used method for statistical visualization of the performance of a classification algorithm, and also a Receiver Operating Characteristic, also known as ROC curve, which is capable of illustrating the performance of a binary classifier as its discrimination threshold is varied. Every classification is marked as a *TP*, *TN*, *FP*, *FN*, and using these values it is possible to calculate some

Figure 7.3: Capture examples for Corridor dataset

---

important measures such as *Precision*, *Recall*, *Accuracy* and *False Positive Rate*, all presented in the form of percentages.

The *Precision* (see Equation 7.1) is the proportion of the predicted positive classifications that were correct, *Recall* (see Equation 7.2), also known as the True Positive Rate (TPR), is the proportion of positive cases that were correctly identified, *Accuracy* (see Equation 7.3) is the proportion of the total number of classifications that were correct, and *False Positive Rate* (FPR) (see Equation 7.4) is the proportion of negative cases that were incorrectly classified as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \tag{7.1}$$

$$\text{Recall (or True Positive Rate)} = \frac{TP}{TP + FN} \times 100 \tag{7.2}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{7.3}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \times 100 \tag{7.4}$$

For each of the tests on the proposed datasets, several combinations of different parameters are used, in order to determine the best combination of values. The most influential parameters for the task of people detection, and therefore the chosen ones to test different values, are the following:

- $\mathcal{W}$ - Width of the line used to create the Distance Transform, in pixels (see section 5.2). Lowering this parameter, to a minimum of value 1, hinders the fitting of the template over the DT, while increasing it facilitates a more perfect fit;

- $\mathcal{S}$ - Minimum confidence level for a ROI to be considered human (see section 5.3). Depending on the quality of the fit of the template over the DT, this score may be higher or lower, with a minimum of 0. Lowering it makes it easier for an object to be considered human, while increasing it makes it harder;
- $\mathcal{C}$ - Number of consecutive detection before classifying a ROI as human (see section 5.3). A great number of FP can be avoided by classifying an object based on more than one detection. Increasing this parameter increases the detection delay, but it also also rewards objects with consistent high confidence.

The combinations of values for each parameter is done by selecting a low, medium and high value adequate to each parameter, based on previous experiments. The values selected for each parameter are as follows: $\mathcal{W} = \{2, 4, 6\}$, $\mathcal{S} = \{0.2, 0.5, 0.8\}$ and $\mathcal{C} = \{1, 10, 20\}$. This combination of parameters are presented in Table 7.1.

Parmeter Set	$\mathcal{W}$	$\mathcal{S}$	$\mathcal{C}$
Set 1	2	0.2	1
Set 2	2	0.2	10
Set 3	2	0.2	20
Set 4	2	0.5	1
Set 5	2	0.5	10
Set 6	2	0.5	20
Set 7	2	0.8	1
Set 8	2	0.8	10
Set 9	2	0.8	20
Set 10	4	0.2	1
Set 11	4	0.2	10
Set 12	4	0.2	20
Set 13	4	0.5	1
Set 14	4	0.5	10
Set 15	4	0.5	20
Set 16	4	0.8	1
Set 17	4	0.8	10
Set 18	4	0.8	20
Set 19	6	0.2	1
Set 20	6	0.2	10
Set 21	6	0.2	20
Set 22	6	0.5	1
Set 23	6	0.5	10
Set 24	6	0.5	20
Set 25	6	0.8	1
Set 26	6	0.8	10
Set 27	6	0.8	20

Table 7.1: Combination of values for each parameter set

The parameter values can be divided in three sub-sets, Set 1 to Set 9, Set 10 to Set 18, and Set 19 to Set 27. This facilitates the interpretation of the results, since, as shown in the

next section, the reaction of the system for each sub-set is similar.

For parameters $\mathcal{W}$ and $\mathcal{S}$, low values enforce a more restrictive classification and higher values allow a more relaxed classification, while for parameter $\mathcal{C}$ the reverse is true. A restrictive classifier reduces the number of incorrect positive classifications but it also increases the number of incorrect negative classifications, while a relaxed classifier generates more positive classifications, but some of these might be incorrect.

As for the hardware on which the system functions, a mid-range laptop is used, with an Intel Core i5-2410M CPU and 4GB of RAM DDR3.

7.3 Test Results

The statistical results presented were obtained by calculating a Confusion Matrix of each combination of dataset and parameter set, and presenting these values in the form of tables (see Table 7.2 and Table 7.3). To facilitate interpretation, charts are also shown: line charts for TP, TN, FP and FN measures (see Figure 7.4 and Figure 7.7), and bar charts for Precision, Recall and Accuracy measures (see Figure 7.5 and Figure 7.8). Furthermore a ROC curve is also presented for each dataset in order to provide a comparisson between the number of FPs and TPs for each parameter set (see Figure 7.6 and Figure 7.9).

Some slight variations can be seen on the total number of classifications for each parameter set, due to the incapacity of the system to process every published frame, and therefore miss some classifications.

7.3.1 Results for Field Dataset

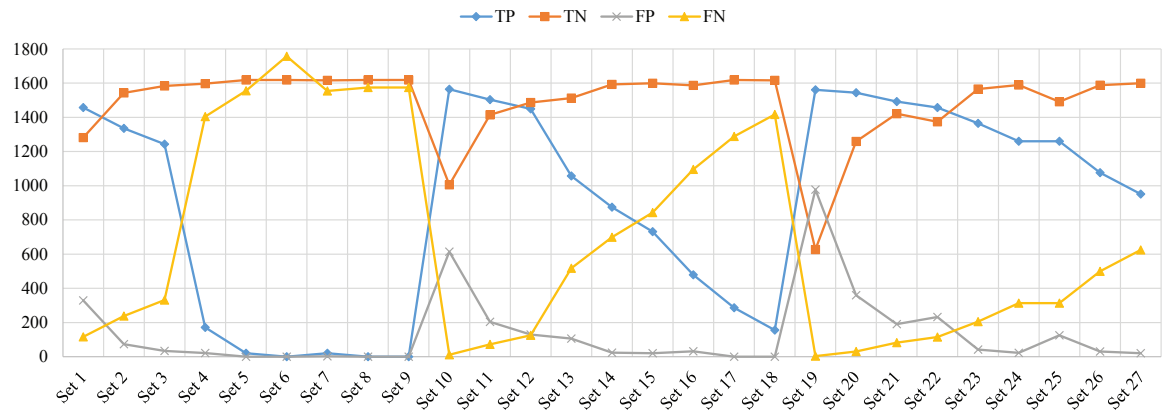


Figure 7.4: Chart for TP, TN, FP and FN measures for Field dataset

Parameter Set	TP	TN	FP	FN	Precision	Recall	Accuracy	FPR
Set 1	1457	1281	329	116	81.58%	92.63%	86.02%	20.43%
Set 2	1336	1543	74	238	94.75%	84.88%	90.22%	4.58%
Set 3	1243	1583	34	331	97.34%	78.97%	88.56%	2.10%
Set 4	171	1597	22	1404	88.60%	10.86%	55.35%	1.36%
Set 5	20	1619	0	1555	100.00%	1.27%	51.32%	0.00%
Set 6	0	1619	0	1757	0.00%	0.00%	50.69%	0.00%
Set 7	21	1616	3	1554	87.50%	1.33%	51.25%	0.19%
Set 8	0	1619	0	1575	0.00%	0.00%	50.69%	0.00%
Set 9	0	1619	0	1575	0.00%	0.00%	50.69%	0.00%
Set 10	1564	1005	614	11	71.81%	99.30%	80.43%	37.92%
Set 11	1503	1415	204	72	88.05%	95.43%	91.36%	12.60%
Set 12	1449	1487	130	125	91.77%	92.06%	92.01%	8.04%
Set 13	1057	1513	106	518	90.89%	67.11%	80.46%	6.55%
Set 14	875	1593	24	699	97.33%	55.59%	77.34%	1.48%
Set 15	732	1599	20	843	97.34%	46.48%	72.98%	1.24%
Set 16	479	1587	32	1096	93.74%	30.41%	64.68%	1.98%
Set 17	287	1619	0	1288	100.00%	18.22%	59.67%	0.00%
Set 18	156	1616	0	1417	100.00%	9.92%	55.57%	0.00%
Set 19	1561	627	977	4	61.51%	99.74%	69.04%	60.91%
Set 20	1544	1259	360	31	81.09%	98.03%	87.76%	22.24%
Set 21	1492	1421	191	82	88.65%	94.79%	91.43%	11.85%
Set 22	1457	1374	232	115	86.26%	92.68%	89.08%	14.45%
Set 23	1365	1565	42	205	97.01%	86.94%	92.23%	2.61%
Set 24	1260	1589	23	314	98.21%	80.05%	89.42%	1.43%
Set 25	1260	1491	125	314	90.97%	80.05%	86.24%	7.74%
Set 26	1076	1588	31	499	97.20%	68.32%	83.41%	1.91%
Set 27	951	1599	20	624	97.94%	60.38%	79.84%	1.24%

Table 7.2: Test results for Field dataset

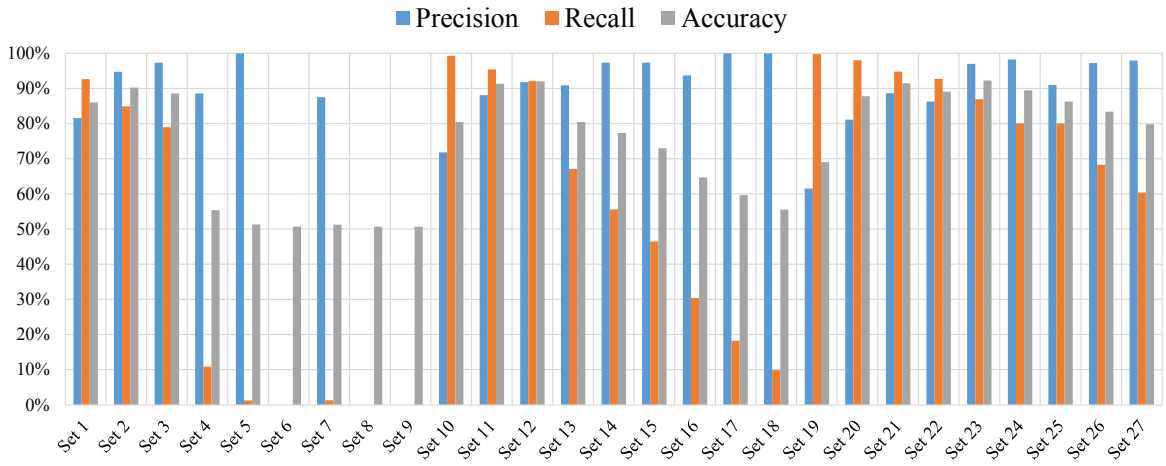


Figure 7.5: Chart for Precision, Recall and Accuracy for Field dataset

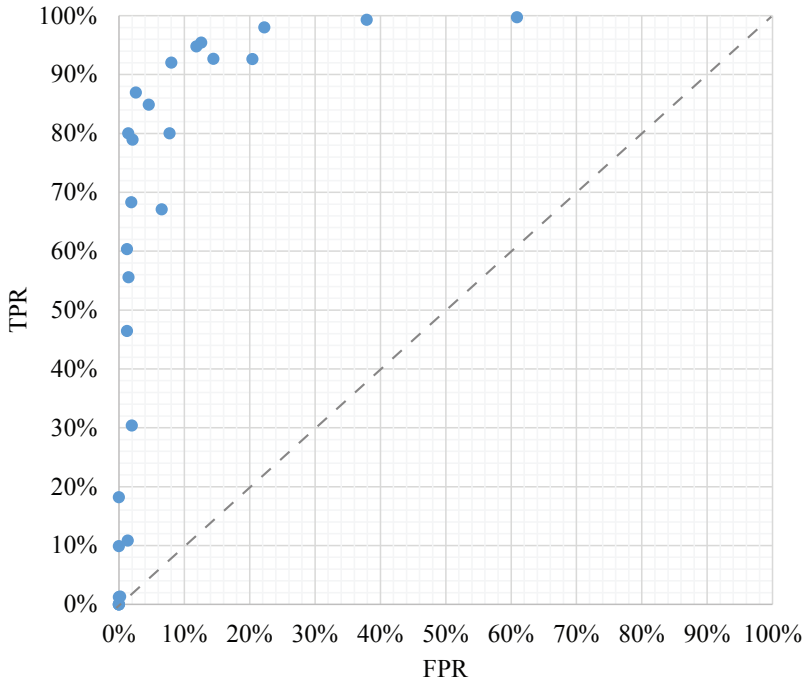


Figure 7.6: ROC curve for dataset Field

7.3.2 Results for Dataset Corridor

Parameter Set	TP	TN	FP	FN	Precision	Recall	Accuracy	FPR
Set 1	1313	2194	1865	112	41.32%	92.14%	63.95%	45.95%
Set 2	1088	3330	735	338	59.68%	76.30%	80.46	18.08%
Set 3	900	3607	452	525	66.57%	63.16%	82.18%	11.14%
Set 4	495	3650	415	931	54.40%	34.71%	75.49%	10.21%
Set 5	270	3968	97	1156	73.57%	18.93%	77.18%	2.39%
Set 6	143	4034	31	1283	82.18%	10.03%	76.07%	0.76%
Set 7	166	3991	74	1260	69.17%	11.64%	75.71%	1.82%
Set 8	94	4065	0	1332	100.00%	6.59	75.74%	0.00%
Set 9	1	4065	0	1425	100.00%	0.07%	74.05%	0.00%
Set 10	1377	1718	2339	49	37.06%	96.56%	56.45%	57.65%
Set 11	1300	3002	1063	126	55.01%	91.16%	78.35%	26.15%
Set 12	1224	3455	610	202	66.74%	85.83%	85.21%	15.01%
Set 13	1088	2935	1130	338	49.05%	76.30%	73.27%	27.80%
Set 14	832	3666	394	593	67.86%	58.39%	82.01%	9.70%
Set 15	630	3877	188	796	77.02%	44.18%	82.08%	4.62%
Set 16	707	3475	590	719	54.51%	49.58%	76.16%	14.51%
Set 17	413	3910	155	1013	72.71%	28.96%	78.73%	3.81%
Set 18	320	4000	65	1106	83.12%	22.44%	78.67%	1.60%
Set 19	1391	1353	2683	31	34.14%	97.82%	50.27%	66.48%
Set 20	1304	2775	1265	118	50.76%	91.70%	74.68%	31.31%
Set 21	1238	3261	804	188	60.63%	86.82%	81.93%	19.78%
Set 22	1291	2341	1703	131	43.12%	90.79%	66.45%	42.11%
Set 23	1049	3362	697	376	60.08%	73.61%	80.43%	17.17%
Set 24	900	3617	442	525	67.06%	63.16%	82.37%	10.89%
Set 25	1153	2783	1247	266	48.04%	81.25%	72.23%	30.94%
Set 26	912	3565	494	513	64.86%	64.00%	81.64%	12.17%
Set 27	771	3742	313	653	71.13%	54.14%	82.37%	7.72%

Table 7.3: Test results for Corridor Dataset

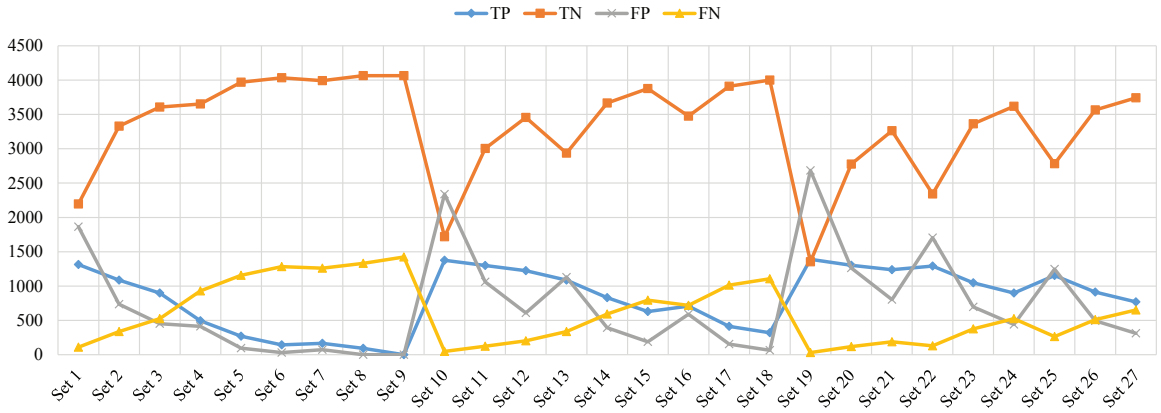


Figure 7.7: Chart for TP, TN, FP and FN measures for dataset Corridor

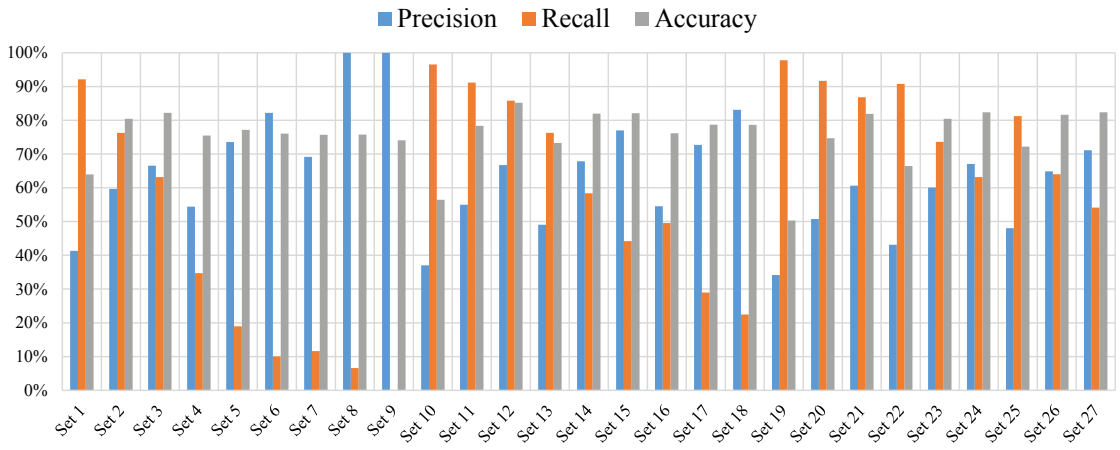


Figure 7.8: Chart for Precision, Recall and Accuracy for dataset Corridor

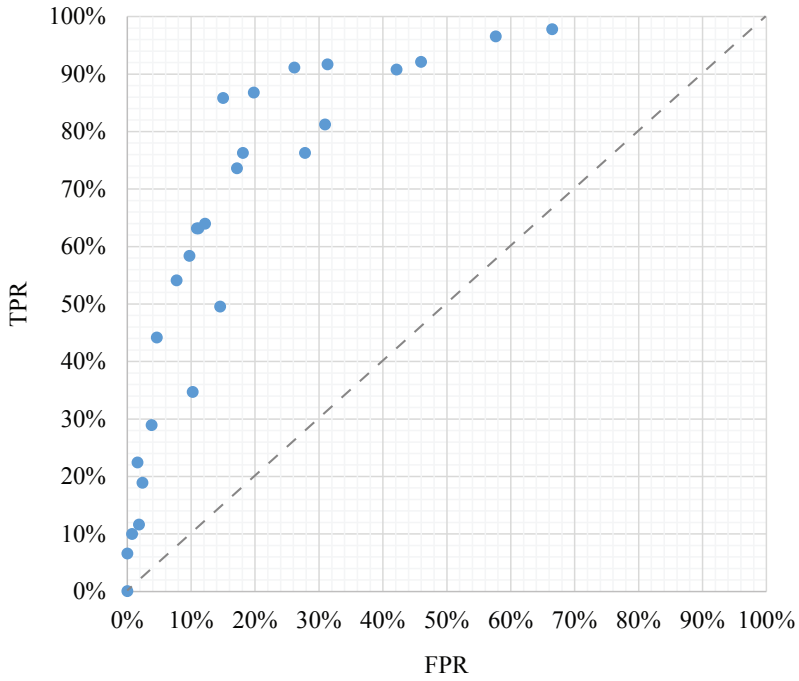


Figure 7.9: ROC curve for dataset Corridor

7.4 Results Analysis

As can be seen in the presented charts and graphs, different parameter sets produce very different results. It is possible to choose different parameter sets depending on the measure that is more adequate to the application. Maximizing all measures, Precision, Recall and Accuracy, with the same parameter set, is not always possible. For cluttered environments, where the number of humans is low when compared to non-human objects, the number of FN will always be lower, raising the Recall of the system. Therefore, in these situations, this metric is not appropriate to characterize the performance of the system. If the number of non-human objects is low when compared to human objects, then FP will be lower, raising the Precision of the system, making the less appropriate measure to characterize the system. The parameter sets described as best for each of the datasets were chosen due to their balanced performance in the following metrics: Precision, Recall and Accuracy.

In each of the three sub-sets of parameter sets it is possible to observe a repetitive behaviour with slight variations, due to the fact that the parameter sets are a combination of different parameters. In order to choose the best parameter values combination, the difference between true and false classifications should be maximized, obtaining the highest possible number of correct classifications while maintaining a low number of incorrect classifications.

Its important to notice that, for each new detection, the number of FN will be increased by the value present in $\mathcal{C}$ since the object will only be classified as a human beyond this number of positive detections.

In the Field dataset, the system's performance presents encouraging results. The parameter sets that present the best results are the ones where $\mathcal{S} = 0.2$, the minimum value for this parameter, and $\mathcal{C} = 20$, the maximum value for this parameter, with any of the values proposed for $\mathcal{W}$. Parameters sets 3, 12 and 21 present the best results, maximizing the difference between true and false classifications (see Figure 7.4), and maximizing the Precision, Recall and Accuracy measures (see Figure 7.5). This indicates that it is more advantageous to maintain a low confidence threshold, while at the same time demanding a continuous classification of ROIs with a confidence higher than the required threshold. For parameter sets composed by a combination of high confidence level and a low number of consecutive detections, the number of TP starts to decrease, while at the same time increasing the number of FN, resulting in lower Recall and Accuracy values, which is undesirable.

The ROC curve shown in Figure 7.6 presents several occurrences near the $(0, 1)$ coordinate, which would indicate a perfect classifier, with no FP occurrences. For the parameter sets that generate points near the top-left corner of the graph, Sets 3, 12 and 21 are among the closest.

A random classifier would generate a line similar to the dashed line that divides the graph of the ROC curve, meaning a *50%* chance of performing a correct classification. The points present in the graph draw a highly angled curve, increasing the area beneath the curve when compared to a random classifier, meaning that, for a given parameter set near the $(0, 1)$ coordinate, the classifier presents a very high Recall, or TPR, and a very low FPR.

For the Field dataset, the parameter set that presents the best results is Set 12, with measures of 91.7% Precision, 92.6% Recall, 91.01% Accuracy and 8.04% False Positive Rate. This parameter set presents slightly better results than Sets 3 and 21 by setting $\mathcal{W} = 4$, which provides enough space to perform the template fitting without being too restrictive, as in Set 3 were the Recall is lower, or too loose, as in Set 21 were the FPR is higher.

Being a more demanding dataset, due to the movement of both the persons and the camera, and due to the natural lighting conditions, the performance of the proposed system

in the Corridor dataset is not as good as on the first dataset. However taking into account that the system is actually capable of performing detection and tracking of people while performing autonomous movement, the results are satisfactory.

As in the first dataset, the method of setting a low threshold for the confidence level ($\mathcal{S} = 0.2$) and a high count of consecutive detections above this threshold ($\mathcal{C} = 20$), also presents the best performance (see Figure 7.7). Sets 12 and 21 present the most satisfactory results, with a slight lead of Set 12 due to higher Precision and lower False Positive Rate.

Comparing the first and second datasets using the same parameter set, Set 12, it is possible to observe a considerably lower Precision, 91.7% to 66.74%, and slightly lower measures for Recall and Accuracy (see Figure 7.8). This is due to the difference in the test environments, which affects the loyalty of the contours in the depth image as to the real contours of the object. In the Field dataset the highest distance that one of the persons present in the dataset achieved is of about six meters, while in the Corridor dataset this distance is increased to nine meters, the maximum depth value at which an object can be segmented for this system, which, in conjunction with the presence of natural lighting, degrades the depth image precision and therefore classification precision.

The ROC curve for the Corridor dataset, shown in Figure 7.9, presents a slight curvature above the diagonal of the graph, and Sets 12 and 21 are among the closest points to the $(0, 1)$ coordinate of the graph, meaning these are the most successful parameters for the classifier. As expect, the curvature of this ROC curve is less noticeable than the ROC curve of the first dataset, demonstrating a worst performance.

By using the ROS middleware it is possible to divide the proposed process in several steps, where each one is associated with independent nodes, that function in different threads. As so, the developed system is based on three different nodes, the Color and Depth Image Preprocessing nodes, which work in parallel, before publishing each processed frame, and the Image Analysis node, which consumes these image frames.

The frequency at which each node is able to operate, as well as the time cost associated with this processing, are presented in Table 7.4.

	Color node		Depth node		Analysis node	
Dataset	Freq. (Hz)	Time (ms)	Freq. (Hz)	Time (ms)	Freq. (Hz)	Time (ms)
Field	29.23	3.17	18.83	49.79	15.26	36.93
Corridor	29.99	3.37	18.98	47.80	16.56	37.02

Table 7.4: Frequency and processing time for each node of the system

Because the Color preprocessing node does not apply any transformation to the color image, and just bypassing each new frame, this node is able to function at nearly the same frequency at which new images are generated, 30 *Hz*. As for the Depth preprocessing node, a set of complex actions are applied, requiring an average of 49 *ms* of processing, allowing for a publishing rate of only 19 *Hz*. The Image Analysis node requires the reception of synchronized frames from the image preprocessing nodes, therefore, the node that publishes at the lowest frequency, between the Color and Depth nodes, will dictate the speed at which this node is able to work. This justifies why the Image Analysis node, even tough it requires less processing time than the Depth preprocessing node, 37 *ms*, is not able to publish at a higher rate.

Being able to operate at 16 *Hz* enables the use of the proposed method in real time systems,

such as the CAMBADA@Home platform. However, because the classification is based on several consecutive frames, performing the same tests as before at the speeds presented in Table 7.4 results in a loss of frames, which degrades the precision of the system.

A comparison has been made with similar systems that use depth images, obtained from different devices, to perform people detection. The results for the proposed system refer to the Field dataset because, as in the datasets used by other researchers, this one presents images obtained from a camera in a fixed position. The results from the works of the following researchers were used: Xia L., et. al. [9], who proposes a model based approach, which detects humans using a 2-D head contour model and a 3-D head surface model; Ikemura, S, et. al. [6], using Relational Depth Similarity Features (RDSF) based on depth information obtained from a TOF camera; Spinello, L., et. al. [3], who took inspiration from the Histogram of Oriented Gradients (HOG) detector to design the Histogram of Oriented Depths (HOD), and the values presented were obtained using the HOD-8, a variation of the authors detector for 8-bit images such as the ones used in the proposed system. It is important to refer that the methods here compared use different datasets, therefore this is not a direct comparison and furthermore some of the measures could not be obtained. In Table 7.5 it is possible to observe that the measures taken with the proposed method are similar to ones stated by other authors, presenting very high Precision, Recall and Accuracy values.

Method	Precision	Recall	Accuracy	Functioning Rate (fps)
Proposed	91%	92%	91%	16
Xia et. al. [9]	100%	96%	98%	*n.a.*
Ikemura et. al. [6]	90%	32%	85%	10
Spinello et. al. [3]	80%	91%	*n.a.*	30

Table 7.5: Comparisson of results against systems proposed by other researchers

Besides the quality of the detection, another important measure is the speed at which the system can function. The proposed system is able to operate at 16 fps on average, on a mid-range laptop with a CPU implementation. As for systems proposed by other researchers, Ikemura, S, et. al., uses a Intel Xeon 3-GHz high end CPU, and operates at 10 fps, while Spinello, L., et. al.'s implementation operates at 30 fps on a high end graphics card, NVidia GTX 480.

Some of the works previously discussed are not present in Table 7.5 because they did not present enough numerical results to perform a comparison or the types of information used are different from depth images. For example, Arras, et al. [17] presents a system with an classification accuracy of 97% using 2D range data; Guan F. et. al. [4] only provides a value of 90% for the detection rate, without regards as to the processing time of his algorithm; Krishnamurthy, S., et. al. [21] uses depth images to perform people detection but only for distances up to three meters, with an 65% detection rate and an exaggerated processing time of 27 seconds per image; Satake, J., et. al. [1] only provides results as to the time required to process each image frame, which is 90 *ms*.

The segmentation process, using histogram analysis proves to be extremely robust, and in the proposed datasets the persons in the captures were always segmented, as well as other objects. As for the correctness of the segmented area, some examples presented overflows when the person is close to other objects. This occurs due to the use of a flood fill algorithm, as discussed in subsection 4.3.2. Therefore, a test was also developed in order to determine the

minimum distance between overlapped objects. Figure 7.10 presents three different captures, with the corresponding color and depth image. The color images shown in *a)*, *c)* and *e)*, are taken from the output of the system, being a mask applied to the regions where humans are not present, and a label is shown over the head of each detected person, presenting information such as the unique ID of the detected person, the confidence level for the current detection and the real world coordinates of the person.

In Figure 7.10, image *a)* shows two human objects correctly detected and segmented, separated by only a few centimetres, side-by-side, and in the corresponding depth image *b)* it is possible to see that there is no overlap between the two persons. The second example, image *c)*, presents a capture with two persons very close, where the person on the left is at 3.56 meters from the Kinect and the person on the right is at 3.09 meters, therefore the distance separating them is of 0.47 meters approximately. In the last example, image *e)*, although the two persons are both unmasked, only one label is shown, indicating that the system considers them to be a single person. Therefore, ROIs who are in contact, for example when one person is in front of the other, and are closer than 0.47 meters approximately, will be segmented as the same object.

Figure 7.10: Captures of humans objects close to each other

Chapter 8

Conclusions

8.1 Conclusion

The main goal of the work under this thesis was the creation of a usable solution for People Detection and Tracking. Furthermore, the use of different types of cameras was also proposed, which due to time constraints, could not be accomplished. Nevertheless a research was made as to the possibility of using a thermal camera to perform people identification.

A solution was achieved and it presents three main stages of processing: detection of ROIs, their classification, and, finally, tracking over the ones classified as human.

In the early stages of the pipeline the depth and color images are processed in order to extract relevant information. The depth image is used to retrieve ROIs through the analysis of its histogram, which, due to the nature of the image, quantifies the mostly occupied areas in the camera's field-of-view, and therefore the most probable locations for a person to be. Using the most occupied levels of the histogram as guides, a slicing of the environment is performed, enabling the discard of less populated areas and therefore increasing the performance of the system. It is important to note that the system is designed to perform detection up to nine meters, while the official recommended play space is up to three meters.

The retrieved ROIs are filtered based on their proportion, area and other features. The ones that pass through these filters are classified as human or not-human using a template matching technique. A novel approach has been taken using the RPROP algorithm, used as a self-localization algorithm for robots, and adapted to this system as template matching algorithm, specialized in people detection. It uses a gradient descent technique to minimize the matching error, and has proven to be extremely useful, as its enhanced computational efficiency enables the matching of multiple templates without a large penalization in terms of processing time. Using multiple templates to perform human detection, not only improves the detection rates, but also allows for the estimation of the person's pose, being the system able to distinguish between 4 poses: facing forward, backward, profile facing left and right.

The system supports simultaneous detections and it is capable of distinguishing between different people present in the same frame. This is possible by generating a histogram from the colors present in the person's silhouette and performing successive comparisons between other persons present in the same capture or even persons previously detected who reappear.

The system was successfully implemented in the CAMBADA@Home robot and, using a mid-range laptop, the system is capable of performing detection and tracking at 16 *Hz*, enabling its use in real-time systems. The tests performed in the proposed datasets indicate

that the system is capable of performing highly-accurate classifications for people detection, when the camera is in a static position. In the second dataset, collected while the CAMBADA@Home performing autonomous movement in an unconstrained environment, it was seen a decrease in the accuracy of the classifications, but it is important to note that most of the detections occurred far away from the camera, where the precision of the depth image is considerably lower.

The CAMBADA@Home project participated in the Free Bots Challenge, a competition at the Robótica 2013 (2013's edition of the Portuguese Robotics Open), where it proved its real-time localization and mapping capabilities while performing people detection in the common area of a high-school. The team was awarded first place. Furthermore, a paper was submitted and accepted in the *XVI Portuguese Conference on Artificial Intelligence*, shown in Appendix A.

In conclusion, the developed system presents characteristics that deem it capable of performing detection in indoor environments, whether they are domestic or professional, such as offices. Using the Kinect camera, the system is able to perform detection for multiple people, in an upright pose, sitting down or standing up and being immune to lighting fluctuations. As for the tracking process the same can not be said because it uses information present in the color image.

8.2 Future Work

Given that this project was fully developed from scratch and will act as a basis for future human-robot applications on the CAMBADA@Home platform, it is important to document some modules that would improve its overall performance, either by completely replacing some of the presented algorithms, or by simply adding functionalities.

The proposed system can be divided in three parts, as indicated by the main chapters of this thesis, attainment of ROIs, object classification and human tracking. The ideas presented in this section are also organized by the area they would improve.

As stated in chapter 7, in order to improve the speed of the overall solution, the speed of the image preprocessing nodes should be increased. One of the ways this can be achieved is by using a more efficient method to recover ROIs from slices, where the current one performs a two-stage process that performs a segmentation of the individual objects present in the slice, and then recovers the ones incorrectly segmented.

The second improvement for this module is related to the method used to detect the limits of an object. The current method requires information on the Kinect's height relative to the ground and inclination. Having another method, independent from this information, should make the system more robust and capable of generating more precise ROIs. Since the depth image only contains information on shapes, and because objects in contact are unavoidably connect by pixels with the same intensity in this image, using the color image to perform this segmentation should prove useful.

As seen in the images present throughout this thesis, the precision of the Kinect depth sensor is not beyond reproach, specially in well lit spaces. Reducing the precision of the depth image from 16 bits to 8 bits certainly reduces the image quality and therefore the precision of the whole system. Hence, a solution should be developed where the image's bit-depth does not have to be reduced and if possible reducing even further the time cost associated with the retrieval of ROIs.

In the classification process of the proposed system, several improvements could also be implemented. Currently the system performs matching over a 2D image, using binary templates. One improvement that would probably increase the precision of the classification, would be to perform matching using, not only the contours of the ROI, but also the values from inside the contour, taking advantage of the information present in the depth image.

The use of templates created using depth images captured from the Kinect, instead of templates proposed by other researchers, increased the precision of the classification system immensely. The system could be further improved if the templates would adapt over time, for each person detected, as in [4], where deformable head-shoulder templates are used. Also, taking inspiration from the method proposed by Mozos, O, et. al. [18], it would be interesting to test the performance of the classification using a multi-part classifier, for different body parts, using templates.

Currently, the discrimination of multiple people is performed by histogram comparison. However, this method is sensitive to changes in illumination, while the person is out of the camera's field-of-view. It should be improved by performing a study over the best color space used to perform the histogram comparison and the weight of each channel. Furthermore, once an ID has been incorrectly assigned, the system is hardly capable of recovering. A study should be made over the use of a multi-hypothesis tracking, used for data association problems. This is a popular method among some researchers and more robust to errors.

The last item for future work is an important one that could not be completed for this thesis, a facial recognition system. The problem of detecting the area of the image where the face is present is solved, as can be seen in section 6.4. Therefore a facial-recognition system, such as the ones covered in section 2.3, should be implemented.

Bibliography

[1] Junji Satake and Jun Miura. Robust Stereo-Based Person Detection and Tracking for a Person Following Robot. 2009.

[2] Mike Andreas Reichinger. Kinect pattern uncovered. 2011.

[3] L. Spinello and K. Arras. People detection in RGB-D data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843. IEEE, 2011.

[4] F. Guan, L. Li, S. Ge, and A. Loh. Robust human detection and identification by using stereo and thermal images in human robot interaction. *International Journal of Information Acquisition*, 04(02):161–183, 2007.

[5] L Spinello and R Siegwart. Human detection using multimodal and multidimensional features. In *In IEEE International Conference on Robotics and Automation*, pages 3264–3269. Ieee, 2008.

[6] Sho Ikemura and Hironobu Fujiyoshi. Real-time human detection using relational depth similarity features. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part IV*, ACCV'10, pages 25–38, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del Solar. Human detection and identification by robots using thermal and visual information in domestic environments. *Journal of Intelligent & Robotic Systems*, 66(1-2):223–243, 2012.

[8] F. Hegger, N. Hochgeschwender, G. Kraetzschmar, and P. Ploeger. People Detection in 3D Point Clouds Using Local Surface Normals. In X. Chen, P. Stone, L. Sucar, and T. Zant, editors, *RoboCup 2012: Robot Soccer World Cup XVI*, volume 7500 of *Lecture Notes in Computer Science*, pages 154–165. Springer Berlin Heidelberg, 2013.

[9] L. Xia, Ch. Chen, and J. Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference*, pages 15–22, 2011.

[10] Robot Operating System Official Website.

[11] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):809–830, 2000.

[12] Stephen J. McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, and Harry Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.

[13] G. Foresti, L. Marcenaro, and C. Regazzoni. Automatic detection and indexing of video-event shots for surveillance applications. *Multimedia, IEEE Transactions on*, 4(4):459–471, 2002.

[14] N. Dalal and W. Triggs. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, 1(3):886–893, 2004.

[15] J. Kovac, P. Peer, and F. Solina. Human skin color clustering for face detection. In *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, volume 2, pages 144–148 vol.2, 2003.

[16] J. Davis and M. Keck. A two-stage template approach to person detection in thermal imagery. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 364–369, 2005.

[17] K.O. Arras, O.M. Mozos, and W. Burgard. Using Boosted Features for the Detection of People in 2D Range Data. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3402–3407, 2007.

[18] O. Mozos, R. Kurazume, and T. Hasegawa. Multi-part people detection using 2d range data. *International Journal of Social Robotics*, 2(1):31–40, 2010.

[19] Luciano Spinello, Kai Oliver Arras, Rudolph Triebel, and Roland Siegwart. A Layered Approach to People Detection in 3D Range Data. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[20] HimanshuPrakash Jain, Anbumani Subramanian, Sukhendu Das, and Anurag Mittal. Real-time upper-body human pose estimation using a depth camera. 6930:227–238, 2011.

[21] Su. Krishnamurthy. Human detection and extraction using kinect depth images. *www1bpt.bridgeport.edu*, 2011.

[22] A. Azarbayejani, C. Wren, and A. Pentland. Real-Time 3-D Tracking of the Human Body], booktitle = Proceedings of IMAGE'COM 96, year = 1996.

[23] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example based object detection in images by components. *IEEE Trans. Pattern Anal. and Machine Intell*, 23:349–361, 2001.

[24] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734–741 vol.2, 2003.

[25] Javier Ruiz-del Solar, Rodrigo Verschae, and Mauricio Correa. Recognition of faces in unconstrained environments: A comparative study. *EURASIP Journal on Advances in Signal Processing*, 2009(iv):1–20, 2009.

[26] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 586–591, 1991.

[27] Javier Ruiz del solar and Pablo Navarrete. Eigenspace-based face recognition: A comparative study of different approaches. *IEEE Trans. Syst., Man, Cybern. C, Cybern*, 35:315–325, 2005.

[28] Mauricio Correa, Gabriel Hermosilla, Rodrigo Verschae, and Javier Ruiz-del Solar. Human detection and identification by robots using thermal and visual information in domestic environments. *Journal of Intelligent & Robotic Systems*, 66(1-2):223–243, 2012.

[29] Gabriel Hermosilla, Patricio Loncomilla, and Javier Ruiz-del Solar. Thermal face recognition using local interest points and descriptors for hri applications. In Javier Ruiz-del Solar, Eric Chown, and PaulG. Plger, editors, *RoboCup 2010: Robot Soccer World Cup XIV*, volume 6556 of *Lecture Notes in Computer Science*, pages 25–35. Springer Berlin Heidelberg, 2011.

[30] Mike Schramm. Kinect: The company behind the tech explains how it works, 2010.

[31] Xenics Gobi-384 Specifications Website.

[32] OpenCV Documentation Website.

[33] S. Suzuki and K. Be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, April 1985.

[34] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.

[35] M. Lauer, S. Lange, and M. Riedmiller. Calculating the perfect match: An efficient and accurate approach for robot self-localization. In *in RoboCup Symposium*, pages 142–153. Springer Verlag, 2005.

[36] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591 vol.1, 1993.

Appendix A

Annexes

Human detection and tracking using a Kinect camera for an autonomous service robot

Luís Ferreira, António Neves, Artur Pereira,
Eurico Pedrosa, and João Cunha

Transverse Activity on Intelligent Robotics,
IEETA/DETI, Universidade de Aveiro
Aveiro, Portugal
{lff,an,artur,efp,joao.cunha}@ua.pt
http://www.ieeta.pt/atri/

Abstract. This paper presents a novel method for people detection and tracking using depth images provided by Kinetic camera. The depth image captured by a Kinect camera is analysed using its histogram, allowing for the depth image to be divided in slices, making the retrieval of regions of interest a simple and computationally light process when compared to point clouds. These regions are then classified as human or not, using a template matching technique. An efficient gradient descent algorithm is used to perform the template matching, using the RPROP algorithm, and the tracking is performed based on color image histogram comparison for each region of interest, in consecutive frames. The proposed method is viable for on-line detection and tracking of people and has been tested in a mobile platform in an unconstrained environment.

Keywords: people detection, people tracking, depth image, template matching, kinect

1 Introduction

In order for a robot to interact with its surroundings it has to be able to do some basic tasks, being one of the most important to see and understand what it is seeing. In recent years, many advances have been made in the Computer Vision research area, where some projects have been deployed and proven to be effective, in the interaction between robots and humans.

The goal of the work presented in this paper is to create an usable solution for people detection and tracking, in an unconstrained environment used in a mobile platform, making future work focused in the interaction between robots and humans a possibility.

The Robot Operating System (ROS) middleware, and C++ in conjunction with the OpenCV framework, were chosen to implement this project. The chosen physical mobile platform was the CAMBADA's team service robot, CAMBADA@Home, built specifically for the @Home challenge present in Robocup competitions and used for academic research.

The remaining of the paper is organized as follows: section 2 presents relevant works used as study cases for this project, section 3 presents an overview of the algorithm and its operation, and section 4 draws some final remarks on the proposed system.

2 Related work

For the past ten years it can be seen an increase of activity in the social robotics research area. The improvement of mobility and processing power of current computers allows for projects like domestic service robots to become more available as time goes by. Two of the key topics in this field are detection and tracking of people.

If one goes back to some of the former works developed on people detection [1], we can see a tendency to use techniques based on background extraction. These can be applied to any type of images (e.g. color, thermal or depth) and present good results on human object detection as long as they fulfil strict requirements (e.g. stationary camera or a model of the background).

The appearing of RGB-D cameras, such as the Microsoft Kinect or Asus Xtion, benefited many projects of the computer vision research area due to the availability of two types of images, color and depth, on the same device while maintaining a very low price when compared to other 3D or thermal devices, such as Laser Range Finders or Long-Wave Infrared (LWIR) cameras. Some related works, such as [2], [3] and [4], use this new type of cameras to perform human detection, identification and tracking.

There are two main approaches preferred by researchers when dealing with human detection. The first is based on machine learning methods, such as AdaBoost [5], [6], [7] or Support Vector Machines, that use features like Histogram of Oriented Gradients (HOG) [8] or Local Surface Normals (LSN) [9] to perform a classification of objects as human or non-human. Other widely used technique is template matching, employed by systems such as [1] or [3], and applied to different types of images.

3 Proposed algorithm

The algorithm presented in this paper makes use of novel methods, some inspired by existing work in the people detection research area, where the most influential is the work developed by Xia, L., et al. [3]. A Kinect camera is used to capture the environment in the form of images, both from the depth camera and the color camera.

The image from the depth camera enables and facilitates the detection of shapes. The process starts by analysing the depth image's histogram in order to detect relevant areas of the image characterized by peeks in the histogram. This enables the slicing of the scene retrieving only part of it in the form of 2D images, that are analysed in order to determine possible regions of interest (ROIs). These ROIs are then classified as human or not using the RPROP algorithm [10],

inspired by its use as a localization method in [11], and now used as part of a template matching technique.

Tracking of ROIs classified as human is performed by histogram comparison on the color image. This technique has been proven to be fast and effective on relating the same ROIs across consecutive frames, even if the regions disappear and reappear due to detection errors, enabling tracking of multiple persons.

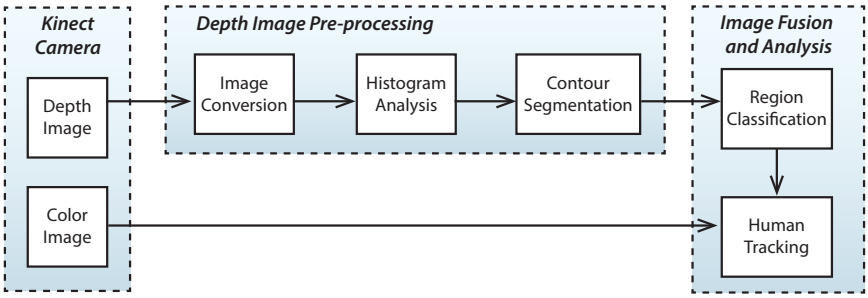


Fig. 1. Overview of the proposed method.

3.1 Obtaining Regions of Interest

The first stage, and essential part of the algorithm, is the detection of regions of interest in the scene. These are obtained performing several operations based only on the depth image image capture by the Kinect camera. In consequent stages of the process these ROIs may be used as masks indicating the regions of the image, both color or depth, that are populated by a possible human object. The images presents throughout the paper show the depth image registered over the color image, meaning that the same object occupies a similar area in both captures.

Depth Image conversion When working with the Kinect camera, image acquisition is an important part of the process, because throughout the proposed method, 8-bit images are used. However, the ROS driver for Kinect used to perform the communication between the camera and the program is only capable of delivering 16-bit images, 65536 different values per pixel, where each pixel carries the distance measured between the plane it is inserted and the camera's plane, in millimetres. Despite being encoded in 16-bits the highest witnessed value measured by the Kinect in different scenes was 9757, meaning the Kinect cannot see beyond 9.7 meters of distance, approximately.

The conversion of the image is justified mainly for two reasons: first, most of the image-processing algorithms available in the OpenCV framework do not support matrices with encoding larger than 8-bits, and so to reduce development time this approach was preferable; second, processing 16-bits images is computationally more costly than processing 8-bit images, and because this project is meant to be applied in a service robot with other algorithms being employed at the same time, such as navigation and localization, computational cost is an important factor.

In order to preserve the information with the possible best precision, when passing from 16-bit images to 8-bit images, the conversion is not applied on the entire original range, from $[0, 65536]$ to $[0, 256]$, but rather on the range relevant for this application. Proceeding in this manner means we are preserving as much precision as possible.

To perform the detection of humans, the template matching algorithm needs a complete vision over the person's head and shoulders. This limits the minimum range at about 1 meter from the Kinect, due to the height that the camera is mounted on the CAMBADA@Home and due to the vertical field of view of the camera. As for the maximum range, the further the object is from the camera, the more irregular its contour will be, and also as stated before the Kinect cannot capture object beyond 9.7, therefore 9 meters was the chosen value for the maximum detection distance. Given this minimum and maximum distances, the conversion is carried out using Equation 1, discarding pixels outside of the relevant range.

$$\mathcal{C}(u,v) = \begin{cases} b\left(\frac{\mathcal{I}(u,v)-\psi\times 1000}{(\gamma-\psi)\times 1000}\right) & \text{, if } \psi < \mathcal{I}(u,v) < \gamma \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

In Equation 1, $\mathcal{C}$ is the matrix representing the converted image and $\mathcal{I}$ is the original image, both the same size (640 columns by 480 rows). Interval $[\psi, \gamma]$ is the relevant depth range in meters, and b the number of bins of the histogram, which in this case is equal to 256 in order to take advantage of the full 8-bit precision.

Histogram analysis Analysing 3D scenes is a computationally heavy task as has been seen by most of the processing time achieved in previous work covered in section 2 of this paper. The method presented in this paper makes use of depth images, instead of point clouds, due to the superior simplicity in data analysis (2D instead of 3D) and use of known image-processing algorithms.

If the image's histogram is calculated, it can effectively demonstrate the most occupied regions in the image. If the objective is then to detect humans in the environment, these may be considered as continuous regions who occupy a portion of the scene. The objective when analysing the histogram is to search for the most occupied regions, represented by mounds in the histogram, and create slices that encompass these, preferably individually.

Due to the conversion made in the previous stage and the interpolation performed natively by the Kinect for distant points, bins higher than a certain value

start to suffer from high variations creating improper local maximums. Therefore, before the detection of the histogram's slices, a median filter is applied only to the bins whose count is equal to 0 to smooth the graph, using Equation 2.

$$\mathcal{H}(i) = \begin{cases} \left(\frac{\mathcal{H}(i-1)+\mathcal{H}(i+1)}{2}\right) & \text{, if } \mathcal{H}(i) = 0 \\ \mathcal{H}(i) & \text{, if } \mathcal{H}(i) > 0 \end{cases} \text{, where } 0 < i < b \tag{2}$$

In Equation 2, $\mathcal{H}$ is the image's histogram array and i the number of the bin, where its value can go from 1 to 255, ignoring bin 0 which is the value assigned to discarded values or pixels outside of the desired conversion range.

The algorithm starts by locating all local maximums, which are characterized by the property represented in Equation 3.

$$\mathcal{M} = \{i : \mathcal{H}(i-1) < \mathcal{H}(i) > \mathcal{H}(i+1)\} \tag{3}$$

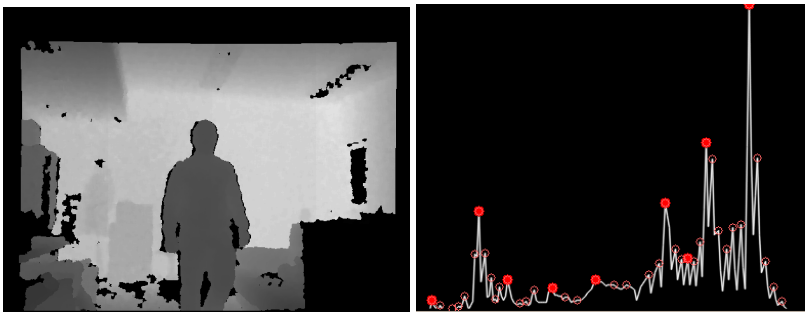
These local maximums, whose indexes are stored in $\mathcal{M}$, represent brightness levels in the scene that are more populated, and if there are any persons (or other objects) in the image, they will most likely be in these levels. Objects in the image have a certain thickness, therefore, determining only the local maximum is not enough, it is necessary to expand these levels into slices that encompass several levels. To optimize the computational cost associated with the processing of each slice, not all maximums will be expanded into a slice, but only the most prominent.

In order to obtain only the most important local maximums, Equation 4 is applied to the previously obtained maximums in $\mathcal{M}$, selecting only second order local maximums, and storing their indexes in $\mathcal{P}$. The second part of the condition was added to reduce even further the number of local maximums selected for expansion, ignoring those who do not stand out from their neighbours with a count higher than ω.

$$\begin{aligned} \mathcal{P} = \{m_i : \mathcal{H}(m_{i-1}) < \mathcal{H}(m_i) > \mathcal{H}(m_{i+1}) \wedge \\ (\mathcal{H}(m_i) - \mathcal{H}(m_{i-1}) > \omega \vee \mathcal{H}(m_i) - \mathcal{H}(m_{i+1}) > \omega)\} \\ \text{, where } m_j = \mathcal{M}(j) \end{aligned} \tag{4}$$

In Figure 2, a capture of the scene is shown in the form of a depth image (*a)*) and its corresponding histogram (*b)*) after applying Equation 2. In the histogram it is possible to discern several peeks and circles over theirs tips. The set of bins selected by Equation 3 are marked as outlined circles, while filled circles mark the bins selected by Equation 4.

Obtaining just the maximums of the histogram is not enough to be able to threshold the image, creating the mentioned 2D slices of the 3D image. In the histogram, large objects, such as boxes or people, can be seen as mounds. To create the slices of the histogram that encompass these mounds, the second order maximums (filled circles in Figure 2 *b)*) are taken as starting points, and in the proposed algorithm the slices are expanded to both sides until the base of the mound is reached. The bases of the mound are characterized by the changing of the growth direction. This means that, starting from the local maximum,



a) b)

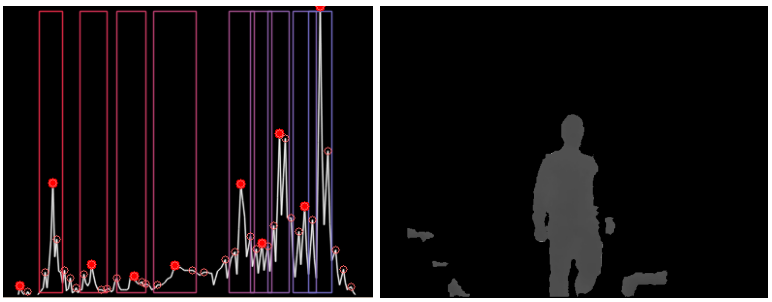
Fig. 2. Example of a captured depth image in a), and its corresponding histogram in b).

consecutive bins in both directions should present a decreasing behaviour (hill sides). When this behaviour changes to increasing it means that the base of the mound has been reached and the slice is complete.

Histograms perform a pixel count for each brightness level. However, there is no information on the position of the pixels. Therefore these slices have to be converted into masks of the real image. This is done by applying a threshold to the image, where pixels that have a brightness encompassed by a given slice of the histogram are marked as 1, and the others are marked as 0. This enables the creation of 2D slices of the 3D image, facilitating the detection of contours, discarding unimportant regions, and thereby reducing computational cost associated with the human detection.

Obtaining Regions of Interest As can be seen in Figure 3, finding the contours of the slices is not enough to create proper ROIs for later classification. A second stage is needed in order to separate independent objects located in the same slice and recover objects that were incorrectly segmented, such as the person in the center whose right arm is missing, in Figure 3 *b)*.

First, the contours of isolated objects in each slice are obtained. Next the minimum bounding box of each contour is calculated, and the centroids of these will be used as a seed pixel for a flood fill algorithm. The lower and upper brightness difference between the seed point and the pixel being flooded are important in order to avoid overflows. These generally happen when two object are in contact, being the most common case, also faced by [3], when the person's feet are in contact with the ground. In the proposed method the preprocessing applied to the depth image reduces its precision and does not employ smoothing techniques that are generally time consuming, therefore Xia, L., et al. solution for this problem does not resolve it. Our proposal is to cut ROIs at floor height or higher, separating the ground from objects that stand on it.



a) b)

Fig. 3. a) Depth image's histogram with slices marked. b) Resulting mask from the first slice.

Using the flood fill method it is possible to restore improperly segmented ROIs during the creation of the image slices, as can be seen in Figure 4. Notice how the arm of the person in Figure 3 b) was missing because it was slightly leaned back, and how using flood fill recovered the complete upper body and enabled the separation of unconnected objects, that were later discarded due to their small size.

a) b) c)

Fig. 4. Color image capture of the scene in a), Human ROI in b) contour of the ROI overlapped in the depth image in c).

Even before regions are classified as human or not, it is possible to discard some of the regions obtained. Before being passed to the classification stage presented in subsection 3.2, all the retrieved ROIs are processed by a filter. This filter is composed by two rules: the first rejects regions based on their width-height ratio, and the second based on their occupied area.

The first rule will not allow regions to have a bounding box with width-height ratio higher than 1.7. Human physiology dictates that the difference between a person's arm span and its height is of a few centimetres Therefore even with the arms wide open a person's proportion should not go beyond 1.0 to 1.2. However, to compensate for occlusion, this limit is extended up to 1.7.

The second rule is that a region cannot have an area bigger than $(96882+\beta)\times e^{(-0.561\times d)}$, where d is the ROIs average depth and β is used as a control value to increase this limit for finner-tunning because a persons size may differ greatly. The equation was obtained by calculating the trend-line for measures taken from a dataset recorded in our lab with few subjects. In both cases, the limits for both proportion and area, should not be very strict because it is preferable for a non-human ROI to pass through this phase, than to incorrectly eliminate human ROIs.

3.2 Information Fusion and Analysis

In the previous stage, regions of interest were retrieved from the depth image. The classification of ROIs, and tracking of the ones classified as human, is performed in this stage, and to do so, information is drawn from the depth and color image. At this point of the project only depth information is used to classify a ROI, however, other methods are being studied to reduce the number of false positives generated by the template matching algorithm, such as the use of a thermal imaging.

Given the ROIs previously obtained, they are classified as human or not through a template matching algorithm. The RPROP algorithm [10] was chosen due to its effectiveness and low computational cost .

In order to keep track of the humans individual position during their presence in the field of view of the camera, a histogram comparison method is used allowing for the same region to be related across consecutive frames, obtaining information from the color image.

Human Classification When using template matching techniques, the choice of the template is crucial for good results, and because the human body presents several degrees of freedom, the choice for which shape to test is done considering the less deformable area. Researchers, such as Xia, L. et al [3] and Krishnamurthy, Su. [4], state that a head-shoulder template, sometimes referred as Ω shape, is the best template to use when trying to detect humans because the head and shoulders are the less deformable part of our body.

Template matching algorithms usually work by sliding a template across an image and calculating an error for the match between the template, and the image being tested. This is a computationally demanding method that requires a lot of processing power if it is intended to run in real-time.

In works, such as [6], [3] and [4], a technique known as *Image Pyramid* is used to perform the template matching for objects with different sizes, due to the perspective effect of monocular cameras caused by the distance of the object relative to the camera. The original image is considered to be the base of the pyramid and at each level the image is downsized. This allows for the same template to be used for detection at different distances.

In our approach the template itself is resized according to the person's distance to the camera and it is tested only over the ROIs, not the entire image.

This allows for a single template to be used for matching at all distances with just one test for each ROI, instead of a number of tests equal to the number of levels of the pyramid. The template is resized according to Equation 5, where s represents the scale factor by which the template will be multiplied, and d is the average depth of the area covered by the ROI in the depth image.

$$s = -52.6 \times log(d) + 130.06 \tag{5}$$

This equation was obtained by fitting the template manually on ROIs classified as human by the system, and obtaining the scale factor necessary for a perfect fit at different depths.

The proposed classification method uses part of the Perfect Match localization algorithm [11], in which the RPROP algorithm [10] searches for the minimal matching error, using a gradient descent technique, in order to fit the template over the ROI's contour. A *Distance Transform* (DT) map is created from the contour of the ROI and the template is then traversed through it, impelled towards the direction that generates the less error. This algorithm is capable of finding a local minimum error for a particular position of the template over the ROI. This position will indicate the center of the head of the person, if it is indeed a human object, with an associated error.

Depending on the starting position, the final and possibly best position of the template can be achieved in 15 iterations, revealing the localization of the person's head. In this particular case of human detection, where humans are assumed to be in an upright position, the algorithm calculates the start position by dividing the ROI's contour in a predefined number of vertical sections, and chooses the one that is most occupied. For example if the person has one arm stretched the head will not be in the middle section of the ROI, but instead more to the left or to the right depending on the arm stretched.

Choosing the best start position is very important due to the possibility of the algorithm to converge to an incorrect local minimum. Figure 5 presents a correct match in b) and an incorrect match in d) due to a local minimum located between the arm and the head.

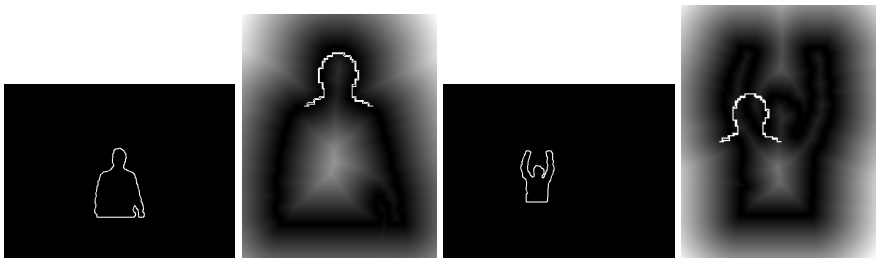


Fig. 5. Example of a correct and an incorrect match location. a) and c) contour of the ROI. b) and d) resulting DT map with the overlapped template.

The classification as human or not is performed by judging the error for the best position. If this error is bellow a certain threshold the region is considered human; if it is above or equal than the region is considered not-human.

Tracking Human Regions In order to keep track of the position of the same person throughout their presence in the scene, it is necessary to relate the same region across consecutive frames. Again, because the human body can shift its shape considerably, depth image is a poor choice when it comes to characterize regions. Consequently tracking relies on the color image provided by the Kinect camera.

A proven detection method is the *Mean Shift* applied to image analysis, in which a particular object is located in a back projection image. Back projection uses the histogram of the desired object as its feature and is then capable of creating an array of statistical probability for the location of that object in a search area.

However, at this point, in addition to the human object contour, its position on the whole image is also known. Therefore, it is not a question of searching for the same object in consecutive frames, but instead to determine whose regions present in the previous frame are still present in the current frame, if any.

Inspired by the method used in the Mean Shift algorithm, the proposed tracking method uses histogram comparison to relate regions across image frames. The HSV color space was chosen for the comparison due to its flexibility when representing colors using only the Hue channel, and its greater robustness to changes in lighting conditions when compared to other color spaces. This is important when tracking a person across different house divisions, where the lighting conditions may change drastically due to different light sources, such as windows or lamps, or even shadows cast by large objects or walls.

The proposed method generates a histogram for the portion of the color image that is masked by ROIs that were classified as human, and normalizes it so that the area occupied by the region is not important but instead the percentage of each color in the ROI. The histograms from the currently tracked ROIs are then compared to previous tracked ROIs in order to calculate a difference value. This difference can be seen as an error that is computed by comparing the value of each histogram's bins, and summing the difference for each channel.

After comparing each current human ROI with previously tracked human ROIs stored in memory, it is necessary to associate them without repetitions. Choosing which current region is assigned to which previous frame's region can be then seen as an optimal assignment problem.

The *Hungarian algorithm* was studied as a possible solution for this problem. However this method did not present a usable solution due to its restrictions. Therefore a new optimal assignment method is proposed where each human ROI selects the region present in the previous frame with the least error, and if two or more ROIs select the same tracked region, only the one with the lowest error will maintain its choice while others will forfeit it and choose a different one, until all regions have chosen different previously tracked regions.

In addition, to ensure that each human object is paired with its equivalent in consecutive frames, the algorithm must also recognize when a error, although being the lowest, is still too high to guarantee that the region is in fact the same. ROIs with an comparison error higher than a certain value are considered new.

4 Final Remarks

In this paper a new method for people detection and tracking is proposed. It is inspired by recent work in this area, but with modifications that allow for a reduced computational cost when compared to other solution that use 3D information.

The method employed for ROIs detection presents satisfactory preliminary results, both in the form of detection rates and computational cost. Slicing the 3D scene in 2D images enables the use of know image analysis techniques while at the same time proves to be a lightweight process in term of computational cost.

The classification phase of the proposed method uses a template matching technique aided by the RPROP gradient descent algorithm, a first use for this algorithm in an image analysis application as far as the authors' concern. Due to the reduced area of the ROIs when compared to the whole image, the algorithm is able to find a solution in 15 iterations and return a position with a local minimum error, which in cases where the head and shoulders are minimally visible is usually the correct location.

By adjusting the minimum error necessary for a region to be considered a human, it is possible to reduce the number of detections incorrectly classified as human, while increasing this threshold value causes human objects to be classified as non-human. Further study is needed in order to determine another classification method that is able to complement these disadvantages, possibly using the color image or even thermal images. Also, because only one template is used for now, people facing sideways to the camera are not properly classified. However, judging by the results of front and back facing people detection, if more templates are considered this problem can be solved without considerably increasing processing time.

Finally, tracking through histogram comparison and association of identical human ROIs across consecutive frames has proven to be effective in most cases, being able to associate the same person during his entire presence in the scene and even after he disappears and reappears, whether it is due to a bad detection in the ROI retrieving phase or simply because the person stepped out of the camera's field of vision. Nevertheless, more tests have to be carried out in order to determine the color space, and individual channel's error weight, that maximize the difference between comparison errors.

The pipeline in mainly divided in two stages, the object detection phase and the classification phase. Object detection is able to extract all ROIs in an image in an average of 48 milliseconds, while object classification and tracking is performed at an average of 54 milliseconds. Because the main stages are im-

plemented in different ROS nodes, the detection stage can be analysing frame i while the classification stage is presenting the results for frame $i - 1$. This makes the system capable of functioning at about 15-20 frames per second in a mid-range laptop with an Intel Core i5 processor, depending on the entropy of the environment.

Acknowledgments

This research is funded by FEDER through the Operational Program Competitiveness Factors - COMPETE and by National Funds through FCT - Foundation for Science and Technology in the context of the project FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011), and Project Cloud Thinking (funded by the QREN Mais Centro program, ref. CENTRO-07-ST24-FEDER-002031)

References

1. I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):809–830, 2000.
2. F. Guan, L. Li, S. Ge, and A. Loh. Robust human detection and identification by using stereo and thermal images in human robot interaction. *International Journal of Information Acquisition*, 04(02):161–183, 2007.
3. L. Xia, Ch. Chen, and J. Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference*, pages 15–22, 2011.
4. Su. Krishnamurthy. Human detection and extraction using kinect depth images. *www1bpt.bridgeport.edu*, 2011.
5. S. Ikemura and H. Fujiyoshi. Real-time human detection using relational depth similarity features. *Computer Vision – ACCV 2010*, pages 1–14, 2011.
6. J. Davis and M. Keck. A two-stage template approach to person detection in thermal imagery. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 364–369, 2005.
7. M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del Solar. Human detection and identification by robots using thermal and visual information in domestic environments. *Journal of Intelligent & Robotic Systems*, 66(1-2):223–243, 2012.
8. L. Spinello and K. Arras. People detection in RGB-D data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843. IEEE, 2011.
9. F. Hegger and N. Hochgeschwender. People Detection in 3D Point Clouds using Local Surface Normals. *ais.uni-bonn.de*, 2011.
10. M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591 vol.1, 1993.
11. M. Lauer, S. Lange, and M. Riedmiller. Calculating the perfect match: An efficient and accurate approach for robot self-localization. In *in RoboCup Symposium*, pages 142–153. Springer Verlag, 2005.